MICROCOMPUTADOR CURSO PRÁTICO

TIRE O MÁXIMO DO MICRO EM 30 SEMANAS

PUBLICAÇÃO SEMANAL ILUSTRADA

Cr$ 6.500

7

Memória em disco a laser • Mapas-k
Micros ajudam deficientes

rioGráfica

MANAUS, ALTAMIRA, MACAPÁ, JI.PARANÁ, PORTO VELHO, RIO BRANCO, SANTARÉM, BOA VISTA, SINOP, ALTA FLORESTA: Cr$ 8.450

MICROCOMPUTADOR CURSO PRÁTICO

APLICAÇÕES

Superando limitações 137

SOFTWARE

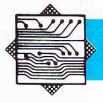
Arquivos na RAM 140

PROJETOS DE PROGRAMAS

Torre de Hanói 142

TÉCNICAS DE PROGRAMAÇÃO

Supersecreto 144



RAIO X

Alfa 3003 146

LINGUAGENS

Quem é o culpado? 148

PERIFÉRICOS

Trilhas espirais 151

CIÊNCIA DA COMPUTAÇÃO

Siga o mapa 154

PERFIL

Música e números 156

TESTE

Depois de ler com atenção todo o fascículo, responda a estas perguntas. Veja a solução no próximo número, junto com um novo teste.

1) Como se chama o sensor óptico controlado por movimentos da cabeça, projetado para que pessoas deficientes possam controlar um micro?
2) Qual a função do registro de identificação num arquivo de dados?
3) Cite uma vantagem e uma desvantagem dos discos a laser, em comparação com as fitas e com os disquetes.
4) Como são gravados os dados num disco a laser?
5) Qual o nome da representação gráfica que usa quadrados preenchidos com os dígitos 0 ou 1 para simbolizar expressões lógicas?

RESPOSTAS DO TESTE ANTERIOR

1) Os dois métodos de acesso aos dados num arquivo aleatório são a indexação e o hashing.
2) Comparando um banco de dados a um fichário, cada ficha seria um registro, e dentro dela o espaço para cada informação (nome, endereço etc.) é um campo.
3) Estes são os três caracteres de controle mais comuns que o micro envia à impressora: retorno do carro (CR, Carriage Return), avanço de linha (LF, Line Feed) e avanço de página (FF, Form Feed).
4) Nas linguagens estruturadas, como o PASCAL, as sub-rotinas podem ser acionadas por um nome, e usam suas próprias variáveis.
5) Aqui está um procedimento em LOGO que mostra uma lista de palavras na ordem inversa:

```
AP INVERTER :LISTAP
  SE :LISTAP=[] ENTAO PARE
  MOSTRE ULTIMO :LISTAP
  INVERTER SEMULTIMO :LISTAP
FIM
```

PLANO DA OBRA

MICROCOMPUTADOR - CURSO PRÁTICO é uma coleção de 30 fascículos de periodicidade semanal. Os 29 primeiros terão, cada um, vinte páginas internas (miolo) e quatro capas, além de uma introdução de dezesseis páginas acompanhando o fascículo 1. O fascículo 30 terá doze páginas e quatro capas, e incluirá o índice geral da obra. Os miolos, encadernados, constituem dois volumes de um curso prático de microcomputação, com seções indicadas por tarjas de diferentes cores.
As duas primeiras capas são descartáveis; as 3.ª e 4.ª trazem programas de jogos. Encadernadas, formam um volume de cem páginas.

COMO ENCADERNAR

As capas duras, incluindo as guardas, estarão à venda simultaneamente com os fascículos 10 (**Volume 1**) e 25 (**Volume 2** e **Jogos**).
Volume 1 - Deve ser encadernado com os elementos dispostos nesta ordem: guardas; frontispício e introdução (publicados com o fascículo 1); miolos dos fascículos de 1 a 15; e guardas.
Volume 2 - Guardas; frontispício (que virá com o fascículo 30); miolos dos fascículos de 16 a 30; e guardas.
Jogos - Guardas; frontispício e sumário (que constituem as 3.ª e 4.ª capas do fascículo 30); 3.ª e 4.ª capas dos fascículos de 1 a 29; e guardas.

Nota: Muitos dos equipamentos que aparecem na seção RAIO X não são normalmente comercializados no Brasil. Sua divulgação é feita a título de informação ao leitor.

rioGráfica **Diretoria Executiva:** Oscar Neves; Filipe Zander; Nilo Sérgio de Almeida; Danilo Esteves Costa; **GRUPO EDUCAÇÃO E CULTURA - Planejamento e Comercialização - Diretor:** Roberto Combochi; **Gerente de Marketing:** Jaime Rodrigues; **Gerente de Produto:** José Mauro Porto; **Gerente do P.C.P.:** Aylton Menezes; **Gerente de Vendas:** Rubens Barbosa; **Coordenadores de Promoção:** Paulo Cesar M. Seixas; Edgar Mello M. Neto; **Gerente de Circulação:** Norberto Martin; **Supervisor de Operações:** Abel Pereira Pinto - **Editorial - Diretor:** João J. Noro; **Editor-chefe:** Maurício Rittner; **Administração Editorial:** Heitor de Souza Paixão; **Editor:** Jesse Navarro; **Revisores:** Isis Augusta Loyolla; Cacilda Guerra; Luiz Vicente Vieira Filho; Marcos Domingos Agathão; Maria da Graça Mendonça Couto; **Diretor de Arte:** Anibal Monteiro; **Chefe de Arte:** Bonifácio Duardes Miranda; **Chefe de Estúdio:** José Yuji Kuribayashi; **Assistente de Arte:** Miguel Luis Escámez Simón. **Colaboradores - Editores-assistentes:** Isa Mara Lando; Luiz Carlos Pizarro Marin; Claudio Marcondes; Nadya Morelo Reis; **Preparação de Texto:** Luiz Carlos Cardoso; **Tradução:** Fernando Dorfman Knijnik; Lúcia Maria Leal Gonçalves; Maria Clara Cescato; Patricia de Paiva e Castro; Regina Amarante; **Pesquisa:** Stella Maria Quentel; **Arte:** Toshio Morobosi; **Assessoria Técnica:** Derval Junqueira de Aquino; Lauro de Lauro; Luiz Henrique Alayon; Luiz Antônio Meloni. As fotos não creditadas pertencem à obra original. Seguem-se os créditos das demais: Fernando Scavone; Jorge's Estúdio. © APSIF Copenhagen, 1984; © Orbis Publishing Ltd., 1984; © Editora Rio Gráfica Ltda., 1985, para a língua portuguesa. **Composição:** ERG Ltda. e AM Produções Gráficas Ltda. **Impressão:** JBIG.

NÚMEROS ATRASADOS - Você poderá comprar os exemplares que faltam em sua coleção, pelo preço do último fascículo posto à venda, em sua banca de jornal preferida. Caso não consiga obtê-los, faça sua solicitação por carta endereçada diretamente à Editora Rio Gráfica Ltda., rua Itapiru, 1209, Rio Comprido, Rio de Janeiro, CEP 20251. O atendimento será feito por via postal. Nas cidades do Rio de Janeiro e de São Paulo, as compras poderão ser efetuadas pessoalmente nos seguintes endereços: Rio de Janeiro - Rua Itapiru, 1209, Rio Comprido; São Paulo - Rua Frei Caneca, 1152, Consolação.

Distribuidor exclusivo para todo o Brasil: Fernando Chinaglia - Rua Teodoro da Silva, 907, Rio de Janeiro. Distribuidor para Portugal: Electroliber Ltda. - Rua Prof. Reinaldo dos Santos, 1488, Lisboa.

SUPERANDO LIMITAÇÕES

Para os deficientes, o micro é bem mais do que uma possibilidade de avanço no aprendizado. Representa, com freqüência, o único meio de superar barreiras entre eles e seu ambiente.

Muitos adultos e crianças necessitam de métodos especiais na educação. Isso devido tanto a problemas físicos, como surdez ou cegueira, quanto a retardo mental. Cada deficiência apresenta problemas específicos, resultantes, muitas vezes, mais do preconceito ou da insensibilidade social do que da deficiência em si.

As escolas especiais são poucas e mal servidas de verbas governamentais. Conseqüentemente, esse não é um mercado lucrativo para os projetistas de hardware e software — desvantagem superada pela dedicação dos voluntários que trabalham nessa área. Um grupo inglês conseguiu angariar o equivalente a quase 20.000 dólares para desenvolver um telefone especial e de baixo custo para surdos. Em muitos países, professores, programadores e engenheiros tomam a iniciativa de desenvolver recursos eletrônicos para o ensino especial.

Na Inglaterra, Centros de Recursos de Microeletrônica para a Educação Especial (Special Education Microeletronics Resources Centres, SEMERC) foram estabelecidos em Manchester, Newcastle e Redbridge para atender a essas necessidades. Uma de suas principais funções — além de avaliar e implantar sistemas — consiste em preparar professores para deficientes e mantê-los informados sobre novidades.

Em alguns casos, é a comunicação que apresenta os maiores problemas. Uma pessoa lúcida e inteligente pode estar fisicamente incapacitada de se comunicar com o mundo exterior. A microtecnologia vem criando canais de comunicação, como o Photonic Wand (''bastão fotônico''), um sensor óptico montado num capacete de plástico e controlado por movimentos da cabeça. Ele torna possível a pessoas com deficiência na fala e no controle motor operar um micro. O bastão conecta-se ao micro pelo canal de entrada analógica. O sensor óptico é semelhante a uma caneta óptica e move um cursor pela tela em resposta a movimentos da cabeça.

Um programa processador de texto chamado Write pode ser usado com o bastão. Ele apresenta o alfabeto na tela, e as letras, maiúsculas ou minúsculas, são escolhidas apontando-se os caracteres. Há uma função de edição e listas de palavras disponíveis. O texto pode ser impresso, gravado ou apagado.

Trabalho pioneiro
Um terminal de trabalho inglês combina uma série de adaptações úteis às pessoas com diferentes deficiências. O Perkins Brailler possibilita a pessoas cegas fornecer informações ao computador e o sintetizador de voz propicia um retorno sonoro. A apresentação do texto na tela foi ampliada para beneficiar os portadores de visão parcial. O terminal de trabalho também se conecta a um teclado conceitual. Embora a foto mostre uma impressora comum, é possível (a preço elevado) a aquisição de impressora especial em Braille.

Outro programa, o Paint, permite desenhar com seis cores, e um terceiro, denominado Music, apresenta na tela um teclado em que as notas são selecionadas com o cursor e tocadas.

O Photonic Wand pode ser usado por pessoas com movimentos instáveis das mãos e mesmo por crianças de apenas oito anos. O SEMERC de Manchester desenvolveu um trabalho com a finalidade de conectar o sensor óptico a um sintetizador de fala. As respostas padronizadas em conversas telefônicas, do tipo ''Repita, por favor'', são selecionadas na tela e então faladas pelo sintetizador. Espera-se ampliar o vocabulário, assim como desenvolver um método simples de conectar o sintetizador ao telefone.

Os computadores são projetados para quem tem a capacidade da visão. Monitores, impres-

soras e plotters são saídas visuais e todos os programas utilizam a tela para apresentar seus dados. A Open University (Universidade Aberta) britânica tem buscado formas de adaptar o hardware disponível e de baixo custo para ser usado por cegos, que colaboram no projeto. Terminais de trabalho, consistindo em um microcomputador BBC, unidade de disco, monitor, impressora, sintetizador de voz, teclado conceitual e um Perkins Brailler adaptado foram instalados em escolas e em outros locais.

O Perkins Brailler foi inventado na década de 40 para datilografia em Braille (código de pontos gravados em relevo no papel, para possibilitar a leitura pelo toque). Ligado ao computador, permite ao cego a leitura da saída impressa. Elaborou-se também um software por meio do qual o computador converte o Braille em texto normal; este pode então ser armazenado, editado ou impresso.

Vários processadores de texto que sintetizam a voz foram desenvolvidos para uso com o teclado padrão. Caracteres digitados e teclas especiais são confirmados por um som. A tecla [Delete] também fala o nome do caractere que foi apagado. O texto é editado utilizando-se um cursor comandado pela voz. Enquanto ele se move na tela, os caracteres, as palavras ou as sentenças são falados. O movimento do cursor pode ser interrompido para adicionar ou apagar partes do texto. Os programas mais sofisticados têm recursos para formatar o texto antes de imprimi-lo. Podem-se definir margens, títulos e espaços para produzir uma cópia final bem-acabada.

A grande vantagem do teclado conceitual sobre o teclado padrão é que ele se adapta a diversas aplicações; pode, por exemplo, ser dividido em quatro seções; uma figura é desenhada em cada quarto, possibilitando o controle de uma tartaruga eletrônica do tipo utilizado pela linguagem LOGO. A pressão em cada uma das diferentes seções envia a tartaruga para a frente, para trás, para a esquerda ou para a direita, o que — no caso de crianças sem capacidade física — é mais simples do que digitar uma instrução. Já foram desenvolvidos um software e uma linguagem, a Starset, para o teclado conceitual.

O mouse usado com o computador Apple Macintosh também evita a "barreira do teclado" e tem amplo potencial no ensino especializado.

Há modernos interruptores que podem ser operados por meios não manuais. Num deles, projetado para pessoas sem movimentos voluntários, dois pequenos discos de metal colocados na pele, próximo aos olhos, detectam o movimento horizontal do globo ocular, e os sinais elétricos decorrentes são amplificados para controlar os interruptores.

No Departamento de Ciência e Sociedade da Universidade de Bradford, uma luz laranja de brilho intermitente foi conectada a uma tartaruga eletrônica para que um jovem deficiente visual pudesse enxergá-la. Usando um teclado conceitual, ele pôde movimentar a tartaruga pelo chão e ver o resultado de suas ações. Para esse rapaz e para muitas crianças deficientes, experiências desse tipo representam um grande passo no sentido de levá-los a participar de seu ambiente, deixando de ser observadores passivos.

Uma especialista inglesa, a dra. Sylvia Weir, apresentou em 1984, numa conferência, notáveis exemplos de progressos alcançados por crianças deficientes físicas e mentais graças ao uso do LOGO e das tartarugas. Um menino autista, de se-

O toque mágico
O Photonic Wand propicia, mesmo a pessoas altamente deficientes, impossibilitadas de falar ou de mover os membros, o controle de um micro. O movimento do bastão é detectado por um sensor óptico e traduzido para um sinal analógico processado pelo computador.

te anos e meio, pôde controlar uma tartaruga por meio de uma caixa com botões, dispositivo semelhante ao teclado conceitual. Ficou tão entusiasmado com a experiência que se pôs a falar pela primeira vez.

A dra. Weir também vem usando o LOGO numa escola especial, com um grupo de adolescentes mentalmente alerta, mas que sofrem de paralisia cerebral. Controlar o computador os ajuda a superar a passividade a eles imposta por sua deficiência.

Um esforço conjunto do SEMERC de Manchester e um grupo local de computação produziu o Micromike, dispositivo para auxiliar crianças com deficiência de fala. Um microfone adaptado, que se conecta ao BBC Micro, possibilita a crianças controlarem várias atividades na tela usando a voz. O software desenvolvido para isso inclui o City, programa com o qual se pode desenhar a silhueta dos edifícios de uma cidade. A altura e a largura dos prédios são determinadas pelo volume e duração da voz. Outros programas permitem o movimento de estrelas no céu, a pilotagem de um barco por corredeiras e o salvamento de um canoeiro por um helicóptero. Eles dão às crianças a oportunidade de aperfeiçoar o controle da voz.

Do mesmo modo que o ensino normal, a educação especial se encontra defasada em relação às conquistas tecnológicas, voltadas mais para áreas administrativas e industriais. A escassez de verbas e de conhecimento especializado retarda o surgimento de hardware e de software para deficientes, e as escolas só lentamente vão adotando a tecnologia da computação. Apesar de tudo, um trabalho fascinante já começa a ser desenvolvido.

Especialmente para tetraplégicos — paralíticos dos quatro membros, que em alguns casos só podem mover olhos e boca —, a Empresa Brasileira de Microinformática (Embramic), de Porto Alegre, instalou um sistema que lhes permite, por meio de sopros apenas, a execução de tarefas antes impossíveis.

Trata-se do Embramic 2000, que reúne um micro Maxxi com unidades de disco, dispositivos eletromecânicos para acionamento e software aplicativo. Soprando num tubo próximo à boca, o deficiente liga o computador e escolhe no monitor de vídeo — em que se desloca um cursor luminoso — a opção desejada: acender luzes, ligar o televisor, abrir portas e discar o telefone. Até princípios de 1985, o Embramic 2000 era o único no gênero no Brasil. Produtos semelhantes, nos Estados Unidos, eram mais onerosos e comandados por voz.

Teclado conceitual
Algumas crianças deficientes não possuem capacidade física para usar o teclado padrão. Um teclado liso e sensível ao toque foi desenvolvido para elas. Chamado de "teclado conceitual", possibilita que películas de acetato substituíveis sejam colocadas sobre ele, adaptando-o a várias aplicações.

E os superdotados?

Crianças superdotadas também têm necessidades especiais de educação e o uso dos micros fornece uma alternativa estimulante para elas, assim como para alguns tipos de deficientes. Muitas crianças com descontrole motor ou com dislexia (dificuldade de ler) são intelectualmente superdotadas.

Nesses casos, o microcomputador oferece um meio de obter controle sobre o ambiente e superar os obstáculos ao desenvolvimento individual, proporcionando à criança superdotada:

- Oportunidade de desenvolver capacidades e interesses em seu próprio ritmo.
- Novos métodos criativos de abordagem de problemas.
- Um meio de comunicação e cooperação com outros estudantes superdotados.
- A oportunidade de praticar exercícios monótonos e cansativos de um modo que mantenha seu interesse.
- Um novo campo para descobertas.

ARQUIVOS NA RAM

Os métodos de manipulação de arquivos tendem a ser específicos para cada tipo de máquina. Como aproveitá-los em micros que só utilizam fitas cassete?

Micros diferentes exigem técnicas distintas para a manipulação de arquivos. Isso torna indispensáveis a adaptação dos programas para cada equipamento e o conhecimento de como a configuração e os comandos próprios de seu sistema se relacionam com os princípios gerais.

Quanto ao armazenamento de arquivos de dados num micro padrão que só utilize cassete, há um primeiro ponto a observar: sistemas desse tipo, por sua própria natureza, não podem lidar com arquivos de acesso aleatório, e os dados são acessados na ordem em que foram armazenados — isto é, seqüencialmente.

A utilização de arquivos seqüenciais envolve a leitura de informações de um arquivo, a manipulação dessas informações e a posterior armazenagem dos dados modificados num segundo arquivo. Assim, devemos ter dois arquivos em uso ("abertos") ao mesmo tempo. Como um gravador cassete não permite passar de uma posição a outra da fita de modo rápido e preciso, esse sistema de "dois arquivos" não é o mais adequado. Constituem exceções alguns sistemas que dispõem de duas entradas para cassete — uma para leitura e outra para gravação.

Por isso, a maioria dos equipamentos disponíveis no nosso mercado só pode acessar um único arquivo seqüencial, o que impõe, na prática, algumas limitações. O arquivo deve ser lido na memória, antes de ser usado; então, se houver modificações a fazer, deve ser novamente gravado no cassete. Procede-se por intervalos, na execução do programa, ou de uma só vez, no seu final. Os arquivos de dados devem ser pequenos o suficiente para caber no espaço da RAM, ainda não utilizado pelo próprio programa. A maior parte dos micros está, dessa forma, restrita a arquivos pouco extensos.

Três métodos principais foram desenvolvidos para armazenagem de informações em cassete. O mais simples não utiliza arquivos de dados separados; ao invés disso, todas as variáveis em uso são armazenadas juntamente com o programa, sempre que o comando SAVE for empregado. Esse método é utilizado pelo TK 82 e também pelo CP 200, da linha Sinclair. Quando se necessita de outro arquivo de dados, nova cópia do programa é usada e gravada com esses dados.

Se tal versão for carregada na memória, os dados serão automaticamente lidos e colocados de volta em suas respectivas variáveis. A vantagem desse método é sua simplicidade — o usuário precisa apenas assegurar-se de que o programa completo seja bem gravado e carregado.

Um sistema mais sofisticado requer um BASIC capaz de armazenar e recuperar matrizes específicas. No Oric Atmos, por exemplo, o comando STORE A$, "NOME" armazena o string A$ na fita, enquanto RECALL A$, "NOME" serve para recuperá-lo. O string inteiro (A$(1), A$(2) etc.) é gravado, apesar de os comandos STORE e RECALL não especificarem seu tamanho. Isso é feito automaticamente quando se dimensiona a matriz no começo do programa.

Esse sistema exige o controle do número de registros da matriz utilizados em seu programa. Uma solução é armazenar a contagem dos registros na matriz antes que ela seja gravada. A maioria dos equipamentos permite um índice zero, de forma que um elemento do tipo A$(0,0) pode ser usado como contador de registros.

O contador de registros será uma variável numérica (em nosso programa usaremos R), mas, se uma matriz de strings estiver sendo usada, ele deverá converter-se numa variável string mediante a seguinte instrução: A$(0,0) = STR$(R). Uma vez recarregada a matriz no computador, reinicializa-se R com: R=VAL(A$(0,0)).

Como muitos micros não possibilitam procedimentos sofisticados de manipulação de arquivos, estes devem ser simulados. Com o arquivo gravado na memória, é fácil usar matrizes em BASIC para tratá-lo como se fosse um arquivo de acesso aleatório.

Vamos supor que uma matriz bidimensional de strings seja usada para armazenar os dados, com uma instrução como DIM A$(100,3). O primeiro índice na matriz designa um registro determinado e o segundo indica um entre quatro campos. Desse modo, os dados são armazenados no formato de tabela — o que equivale a um arquivo de acesso aleatório.

Outro recurso encontrado em algumas máquinas é o comando APPEND. Ele permite acrescentar dados no final de um arquivo seqüencial sem que seja necessário lê-lo todo para criar uma nova versão. Certos computadores possuem um comando para se pular determinado número de registros de um arquivo seqüencial e, desse modo, simular um arquivo de acesso aleatório.

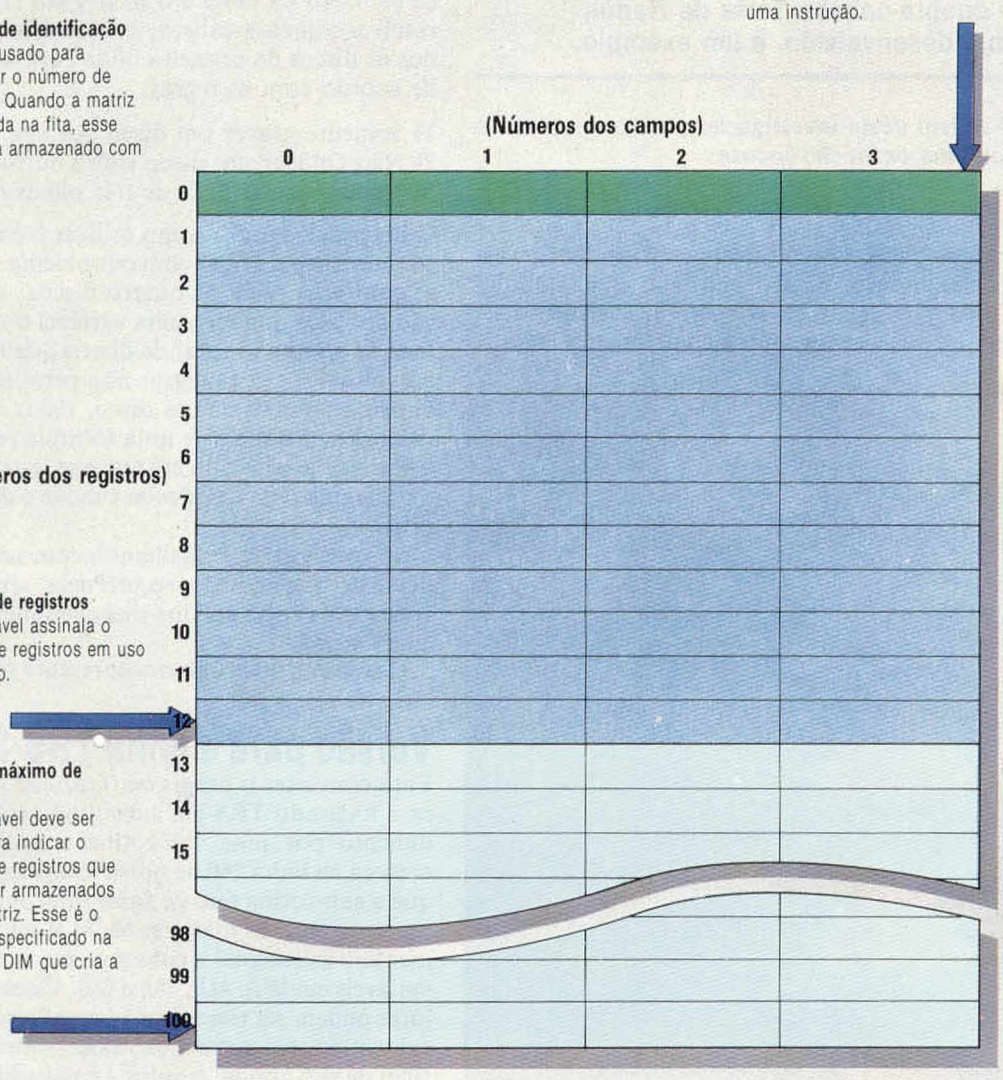
Embora a manipulação de arquivos dependa muito do tipo de equipamento, os princípios gerais são sempre os mesmos, não importando o micro utilizado.

Como armazenar registros e campos numa matriz em BASIC

Uma matriz bidimensional de strings pode ser usada para simular um arquivo de acesso aleatório. O primeiro índice identifica um registro específico e o segundo, os campos dentro do registro.

Número de campos
É bastante útil ter esse número definido numa variável no começo do programa. Isso facilita a mudança posterior para registros maiores, porque significa alterar apenas uma instrução.

Registro de identificação
A$(0,0) é usado para armazenar o número de registros. Quando a matriz for gravada na fita, esse valor será armazenado com ela.



Número de registros
Uma variável assinala o número de registros em uso no arquivo.

Número máximo de registros
Uma variável deve ser usada para indicar o número de registros que podem ser armazenados numa matriz. Esse é o número especificado na instrução DIM que cria a matriz.

Eliminação de um registro
Este programa remove da matriz o registro número N. Todos os registros abaixo deste são movidos para uma posição mais acima, anulando, assim, os dados anteriormente armazenados na posição N.

```
100 LET R=R + 1
110 IF > M THEN PRINT "MATRIZ
CHEIA":RETURN
120 FOR I=R TO N + STEP - 1
130 FOR J=0 TO F
140 LET A$(I,J)=A$(I - 1,J)
150 NEXT J: NEXT I
170 LET A$(N,0)=N$: LET A$(N,1)=C$
180 LET A$(N,2)=D$: LET A$(N,3)=E$
190 RETURN
```

Inserção de um registro
Este programa acrescenta um novo registro na matriz, na posição N. Todos os registros além do ponto da inserção são deslocados para uma posição mais abaixo, criando assim o espaço necessário para os novos dados.

```
200 FOR I=N TO R - 1
210 FOR J=0 TO F
220 LET A$(I,J)=A$(I + 1,J)
230 NEXT J: NEXT I
240 LET R=R - 1
250 RETURN
```

TORRE DE HANÓI

O conhecimento da recorrência, técnica usada na programação avançada, pode acrescentar novas dimensões a programas em BASIC. O quebra-cabeça Torre de Hanói, aqui desenvolvido, é um exemplo.

O objeto desta investigação pode ser resumido por uma definição jocosa:

Recorrência: *ver* Recorrência

Essa definição circular demonstra um aspecto essencial da recorrência — ou seja, algo que é definido por meio de si mesmo. Deixou-se de considerar, porém, outra característica importante: para uma recorrência (ou recursion) ser operável, deve haver uma saída da circularidade.

O quebra-cabeça Torre de Hanói, que ilustra a recorrência, consiste numa pilha de discos dispostos por ordem de tamanho: o disco maior fica embaixo da pilha e o menor em cima. Para resolver o quebra-cabeça, é necessário passar todos os discos da primeira pilha para a segunda, de acordo com as regras:

1) Somente mover um disco por vez.
2) Não colocar um disco sobre outro menor.
3) Nunca formar mais de três pilhas de discos.

O diagrama mostra como utilizar o conceito de recorrência para lidar com o problema. Começa-se com uma pilha de quatro discos. A atribuição do valor quatro a uma variável n serve para indicar o número total de discos que devem ser movidos. Como as regras não permitem o movimento de mais de um disco, reduz-se em 1 o valor de n por meio de uma fórmula recorrente. Continua-se o cálculo até que n se iguale a 1. Aí, o programa pára de calcular e move o disco apropriado.

Se você estiver trabalhando com uma versão de BASIC que admite a recorrência, será fácil escrever um programa que siga exatamente o processo acima.

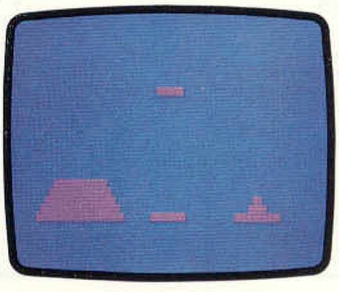
O restante do programa apresenta as imagens na tela.

Mova um disco

O programa pede ao jogador que digite o número de discos que serão movidos. Esse número é colocado na pilha A. Em seguida, o algoritmo recorrente move um disco por vez, formando as pilhas B e C, nos três primeiros estágios. Por fim, a pilha inicial de n discos é reduzida a zero: não há mais discos para mover, como mostra o resultado final. Ao término dos movimentos, n discos estarão reposicionados em ordem seqüencial na pilha B.

Versão para a linha TRS-80

Para converter o programa Torre de Hanói para o BASIC do TRS-80, substituímos um procedimento por uma sub-rotina recorrente, que começa na linha 260 de nossa listagem. Cada vez que a sub-rotina tem de fazer uma chamada recorrente para as matrizes M, A, B ou C, ela aumenta o contador J e coloca os novos valores de variáveis em M(J), A(J), B(J) e C(J). Esses novos valores podem ser usados na chamada seguinte da sub-rotina sem mexer nos valores anteriores. No final da sub-rotina, o valor J é reduzido, restaurando assim os antigos valores. Esse método sempre pode ser usado para escrever sub-rotinas recorrentes em BASIC, por mais complexa que seja a recorrência.



Problemas recorrentes
Assim aparece o jogo Torre de Hanói ao ser rodado no micro Spectrum, da Sinclair. Se você tentar acompanhar a resolução do problema por este micro, preste muita atenção, pois ele é bem rápido!

Linha_TRS-80

```
100 REM*************************************
120 REM*          PARA LINHA TRS-80          *
130 REM*           TORRE DE  HANOI           *
150 REM*************************************
170 CLS: CLEAR 1000: DIM M(100), A(100), B(100), C(100)
180 DIM D$(10), H(3), P(3,10)
190 GOSUB 690
200 PRINT@0,"";:INPUT"QUANTOS DISCOS ";N
210 IF N<1 OR N>10 THEN GOTO 200
220 GOSUB 620
230 J=1: M(J)=N: A(J)=1: B(J)=2: C(J)=3
240 GOSUB 260
250 GOTO 200
260 IF M(J)=1 THEN GOSUB 450  :RETURN
270 J=J+1
280 M(J)=M(J-1)-1
290 A(J)=A(J-1)
300 B(J)=C(J-1)
310 C(J)=B(J-1)
320 GOSUB 260
330 M(J)=1
340 A(J)=A(J-1)
350 B(J)=B(J-1)
360 C(J)=C(J-1)
370 GOSUB 260
380 M(J)=M(J-1)-1
390 A(J)=C(J-1)
400 B(J)=B(J-1)
410 C(J)=A(J-1)
420 GOSUB 260
430 J=J-1
440 RETURN
450 PA=A(J): PB=B(J)
460 M$=D$(P(PA,N+1-H(PA)))
470 FOR I=14-H(PA) TO 2 STEP -1
480 PRINT@(I-1)*64+15*(PA-1),M$;
490 PRINT@I*64+15*(PA-1),B$;
500 NEXTI
510 FOR I=15*(PA-1) TO 15*(PB-1) STEP SGN(PB-PA)
520 PRINT@1*64+I,M$;
530 PRINT@1*64+I,B$;
540 NEXTI
550 FOR I=1 TO 12-H(PB)
560 PRINT@I*64+15*(PB-1),B$;
570 PRINT@(I+1)*64+15*(PB-1),M$;
580 NEXTI
590 H(PB)=H(PB)+1: P(PB,N+1-H(PB))=P(PA,N+1-H(PA))
600 P(PA,N+1-H(PA))=0: H(PA)=H(PA)-1
610 RETURN
620 CLS
630 FORI=1 TO N
640 PRINT@(13-N+I)*64,D$(I);
650 P(1,I)=I: P(2,I)=0: P(3,I)=0
660 NEXTI
670 H(1)=N: H(2)=0: H(3)=0
680 RETURN
690 FORI=1TO9STEP2
700 D$(I)=STRING$(5-INT(I/2)," ")+CHR$(170)+STRING$(2*INT(I/2)
    ,191)+CHR$(149)+STRING$(5-INT(I/2)," ")
710 D$(I+1)=STRING$(5-INT(I/2)," ")+STRING$(I+1,191)+STRING$
    (5-INT(I/2)," ")
720 NEXTI
730 B$=STRING$(12," ")
740 RETURN
```

SUPERSECRETO

Os programas podem ser construídos a partir de blocos independentes chamados "módulos". Veremos, a seguir, como utilizá-los no desenvolvimento de um programa completo.

Quando se constrói um programa, é boa idéia desenvolver uma estrutura genérica. Nesta, as rotinas e aplicação geral podem ser utilizadas por outras, de especialização crescente, em níveis mais altos — todas sob a direção de um único módulo de controle no topo. Essa estrutura piramidal permite-nos usar um método de programação chamado "refinamento" ou "projeto topo-base".

O primeiro passo é a elaboração do programa de controle, localizado no topo. Descrevemos suas funções em termos de chamadas a rotinas de nível "inferior"; em seguida, passamos para o nível de baixo e descrevemos o trabalho de cada uma das rotinas chamadas pelo módulo do topo. Cada rotina é descrita em termos das rotinas que ela deve chamar; o processo é repetido até que se alcance o nível mais baixo. Nesse estágio, as funções desempenhadas pela rotina que estamos descrevendo são tão simples que podem ser definidas com o uso da própria linguagem de programação.

Como exemplo, vejamos o projeto de um jogo de forca. Em vez de o jogador adivinhar uma palavra escolhida pelo programa, como na maioria das versões para computador desse jogo, queremos um programa que adivinhe uma palavra escolhida por nós. Um modo de conseguir isso sem dar ao programa uma longa lista de palavras é fornecer dados sobre a probabilidade da ocorrência de determinadas seqüências de letras.

```
100 REM INICIALIZAR VARIAVEIS E MATRIZES
500 REM ****** ROTINA DE CONTROLE ******
510 REM
520 GOSUB 1000 : REM TITULO E TELAS DE
    AJUDA
530 GOSUB 2000 : REM MONTA A TELA
540 GOSUB 4000 : REM LE O TAMANHO DA
    PALAVRA
550 GOSUB 8000 : REM CARREGA O ARQUIVO
    DE PALAVRAS APROPRIADO
560 GOSUB 3000 : TENTA UMA LETRA
570 GOSUB 4500 : REM VERIFICA SE A
    TENTATIVA FOI CORRETA
580 GOSUB 5000 : REM ATUALIZA A TELA
590 IF JOGO_NAO_TERMINADO THEN 560 :
    REM CONTINUA FAZENDO TENTATIVAS ATE
    O FINAL DO JOGO
600 IF CORRETO THEN GOSUB 10000 ELSE
    GOSUB 11000 : REM FINALIZA DE ACORDO
    COM O RESULTADO
610 GOSUB 6000 : REM PERGUNTA SE O
    JOGADOR QUER JOGAR DE NOVO
620 IF SIM THEN 530 : REM INICIA DE NOVO
630 GOSUB 7000 : REM TCHAU E FIM
640 END
```

Sabe-se de antemão que algumas coisas devem ser feitas: é preciso definir variáveis, dimensionar matrizes, diagramar e atualizar a tela do jogo. Além disso, é necessário escrever rotinas que contem pontos, sugiram letras e, por fim, encerrem o jogo.

Vamos partir de uma instrução REM simples para indicar que será preciso definir variáveis e matrizes — deixando para completar os detalhes mais tarde. A rotina de controle propriamente dita é apenas um par de loops. O loop externo (linha 620) verifica se o jogador quer finalizar a sessão, enquanto o loop interno (linha 590) testa se o jogo terminou.

Para testar a rotina de controle, criam-se subrotinas simuladas correspondentes aos GOSUBs. Cada GOSUB deve ter uma linha REM para explicar sua função e deve começar numa linha com número conveniente — de preferência um número "redondo", como 1000 ou 5000. Rotinas com funções similares devem ter seus números de linhas padronizados; isso vai facilitar muito quando as rotinas forem transportadas de um programa para outro. Por exemplo, instruções de jogo podem estar sempre contidas numa sub-rotina que comece na linha 1000.

A rotina de controle é curta e simples. Cabe inteira na tela: isso torna mais fácil eliminar eventuais problemas, em comparação com programas que se estendam por várias telas. As variáveis JOGO NAO TERMINADO, CORRETO e SIM são sinalizações (flags) colocadas nas várias subrotinas chamadas pela rotina de controle; são usadas para determinar se o programa de controle funciona corretamente. É muito fácil detectar qualquer erro de lógica nessa rotina de controle tão simples.

Nesse ponto, deve-se examinar a estrutura do programa, para ver se este se comporta da maneira esperada em todas as circunstâncias. Pode-se, também, melhorar o projeto do programa. Deixar, por exemplo, as instruções acessíveis em qualquer estágio do jogo, manter um registro de quantos jogos o computador ou o jogador ganharam, ou, ainda, listar as palavras que derrotaram o computador. Todas essas modificações podem ser feitas nesse estágio.

A etapa seguinte consiste na especificação de cada uma das rotinas chamadas pelo programa de controle. Nossas listagens sugerem duas des-

sas rotinas. A primeira, que começa na linha 4000, pede ao usuário um número entre 1 e 20 (o comprimento da palavra). Ela usa uma sub-rotina de aplicação geral que se presume existir na linha 51000, a qual toma uma variável alfanumérica (string) especificada em PROMPT$, a imprime e aceita um número inteiro ao qual o usuário dá entrada. Se esse número não estiver entre os limites definidos por MIN% e MAX%, surgirá uma mensagem de erro e será pedido ao usuário que entre com um novo número. Essa sub-rotina pode facilmente ser usada em outros programas; recomenda-se que o usuário monte uma biblioteca desses módulos genéricos.

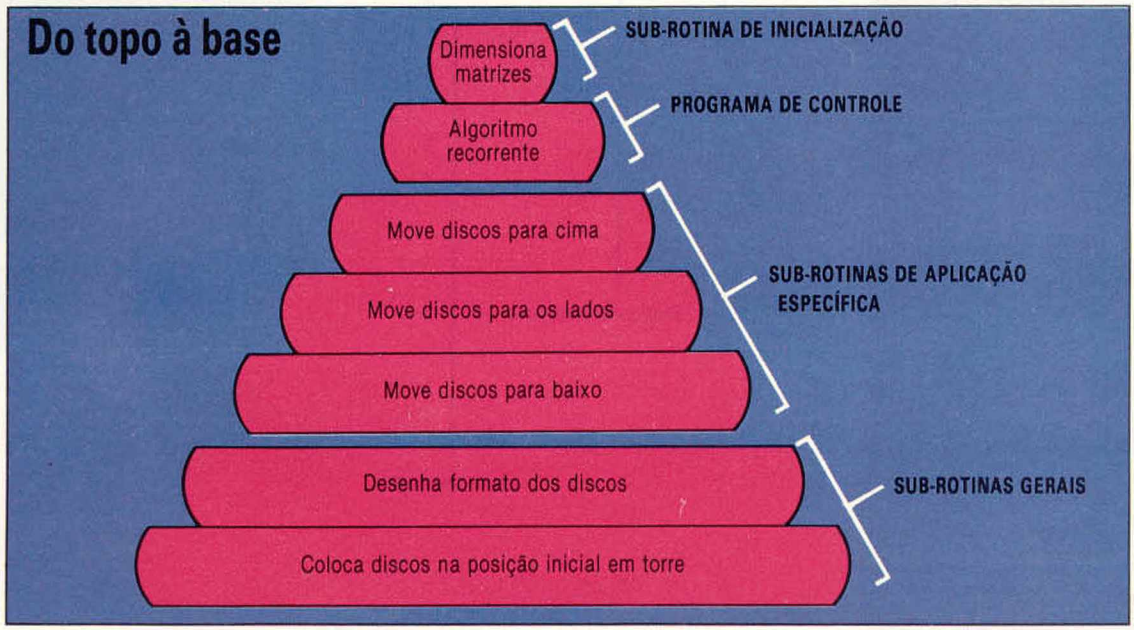
```
4000 REM PEDE AO JOGADOR O TAMANHO DA
     PALAVRA
4010 REM
4020 PROMPT$ = "QUANTAS LETRAS TEM A SUA
     PALAVRA ?"
4030 MIN% = 1
4040 MAX% = 20
4050 GOSUB 51000 : REM ACEITA UM NUMERO
     INTEIRO ENTRE MIN% E MAX%
4060 COMPR_L% = RESP% : REM RESP% E A
     RESPOSTA USADA PELA SUBROTINA NA
     LINHA 51000
4070 RETURN
8000 REM CARREGA ARQUIVO DE PALAVRAS
8010 REM
8020 IF COMPR_L% > 7 THEN ARQ_L% = 8
     ELSE ARQ_L% = COMPR_L%
8030 ARQNUM_L$ = STR$(ARQ_L%)
8040 ARQNOME$ = "TABELA" + ARQNUM_L$
8050 GOSUB 9000 : REM OPEN, READ E CLOSE
     O ARQUIVO DE PALAVRAS COM O COMPRI-
     MENTO APROPRIADO
8060 RETURN
```

A outra rotina, que começa na linha 8000, usa variáveis locais (ARQ L% e ARQNUM L$). Assumimos que os dados necessários para se adivinhar uma letra estão, por exemplo, em oito conjuntos de tabelas. Como só queremos um conjunto de dados na RAM de cada vez, devemos construir um string ARQNOME$ para guardar o nome do arquivo de dados e então chamar a sub-rotina na linha 9000 para que o leia.

Em muitos casos, percebe-se que o programa se move diretamente de uma rotina para outra. No entanto, pode-se criar uma rotina extra para chamar sucessivamente cada uma das outras. Isso pode parecer desnecessário, mas permite que se controle o fluxo do programa e apresenta ainda a vantagem de manter os módulos separados, de modo que podem ser acrescentados a outros programas.

Um recurso para facilitar o projeto de sub-rotinas "versáteis", transportáveis de um programa para outro, é a substituição de constantes por variáveis. É também importante que todas as sub-rotinas sejam bem documentadas. A documentação deve especificar a aplicação exata da rotina, dando detalhes das variáveis usadas, dos valores esperados para a entrada e a saída e de quaisquer efeitos secundários — deslocamentos do cursor, mudança do mapa da memória, fechamento de arquivo etc.

Uma codificação padronizada também facilita bastante; certifique-se de que todos os números de linha obedecem a um intervalo fixo, de que os títulos e comentários estão restritos a um dado número de linhas no começo da rotina e de que o RETURN está sempre na última linha. Não esqueça de anotar o número da primeira e da última linha de cada rotina. Quando for necessária uma rotina de sua biblioteca, confirme se o programa tem um espaço apropriado em seus números de linha e então inclua a sub-rotina no programa com um comando MERGE. Se o seu micro não tiver o comando MERGE, poderá ser usado um editor de texto para combinar programas gravados no formato ASCII em vez da forma "simbólica" usual. Se isso não for possível, as sub-rotinas da sua biblioteca deverão ser digitadas cada vez que forem usadas. Contudo, o fato de não precisarem ser elaboradas já justifica o trabalho extra.



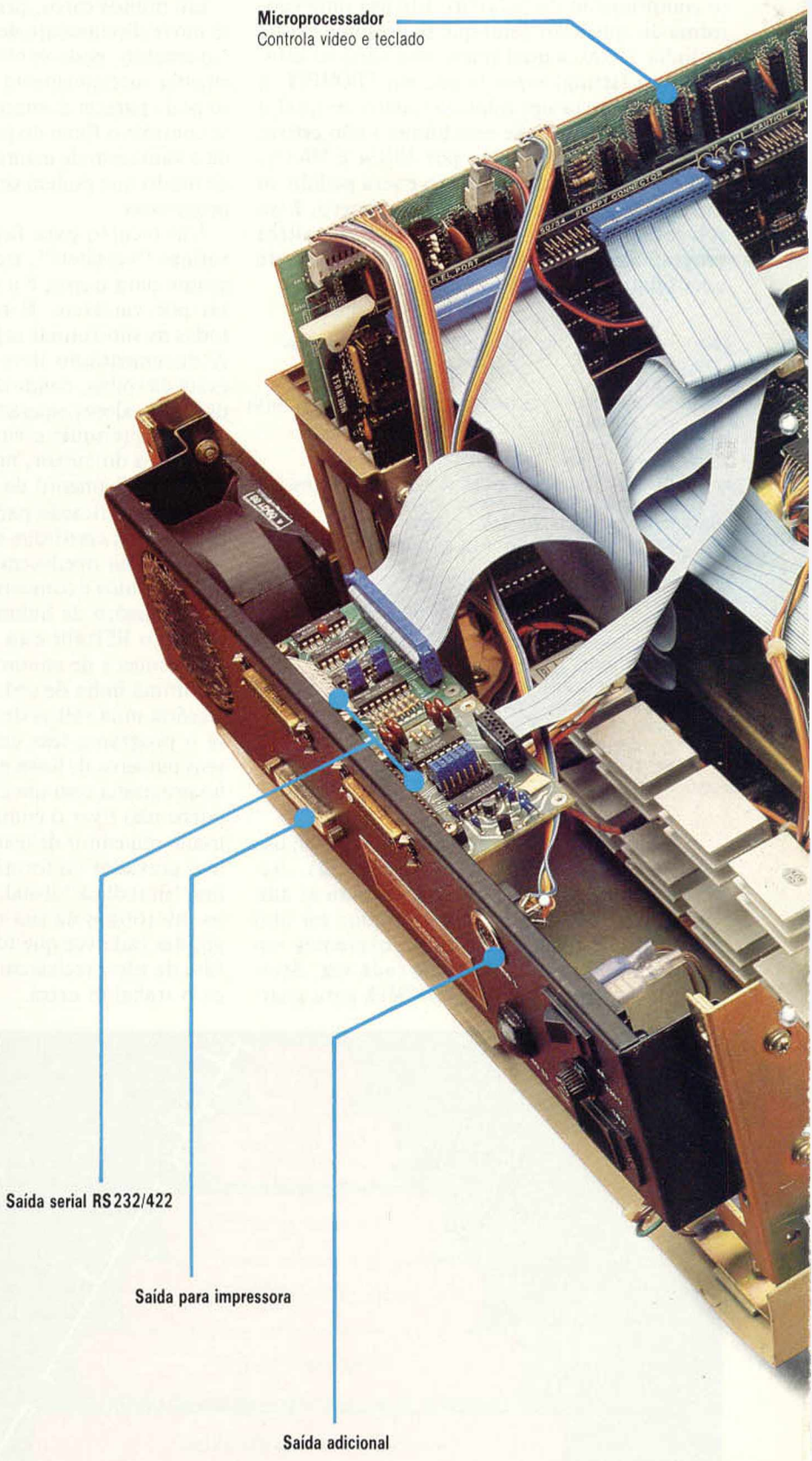
Programação topo-base
O diagrama ilustra o princípio da programação topo-base. É o programa da Torre de Hanói, mostrado anteriormente. O nível mais alto da estrutura representa o programa de definição, que deve ser completado antes de se executar o resto do programa. O programa de controle, no diagrama, representa o algoritmo recorrente, que efetua os cálculos e, se preciso, chama as outras sub-rotinas. As sub-rotinas de aplicação específica são usadas para movimentar os discos de uma pilha para outra na tela. As duas últimas seções do diagrama, sub-rotinas gerais, representam as duas seções finais do programa e são usadas para formatar a tela inicial e criar o desenho dos discos. Compare esse diagrama com a listagem e veja que o programa possui a mesma estrutura.

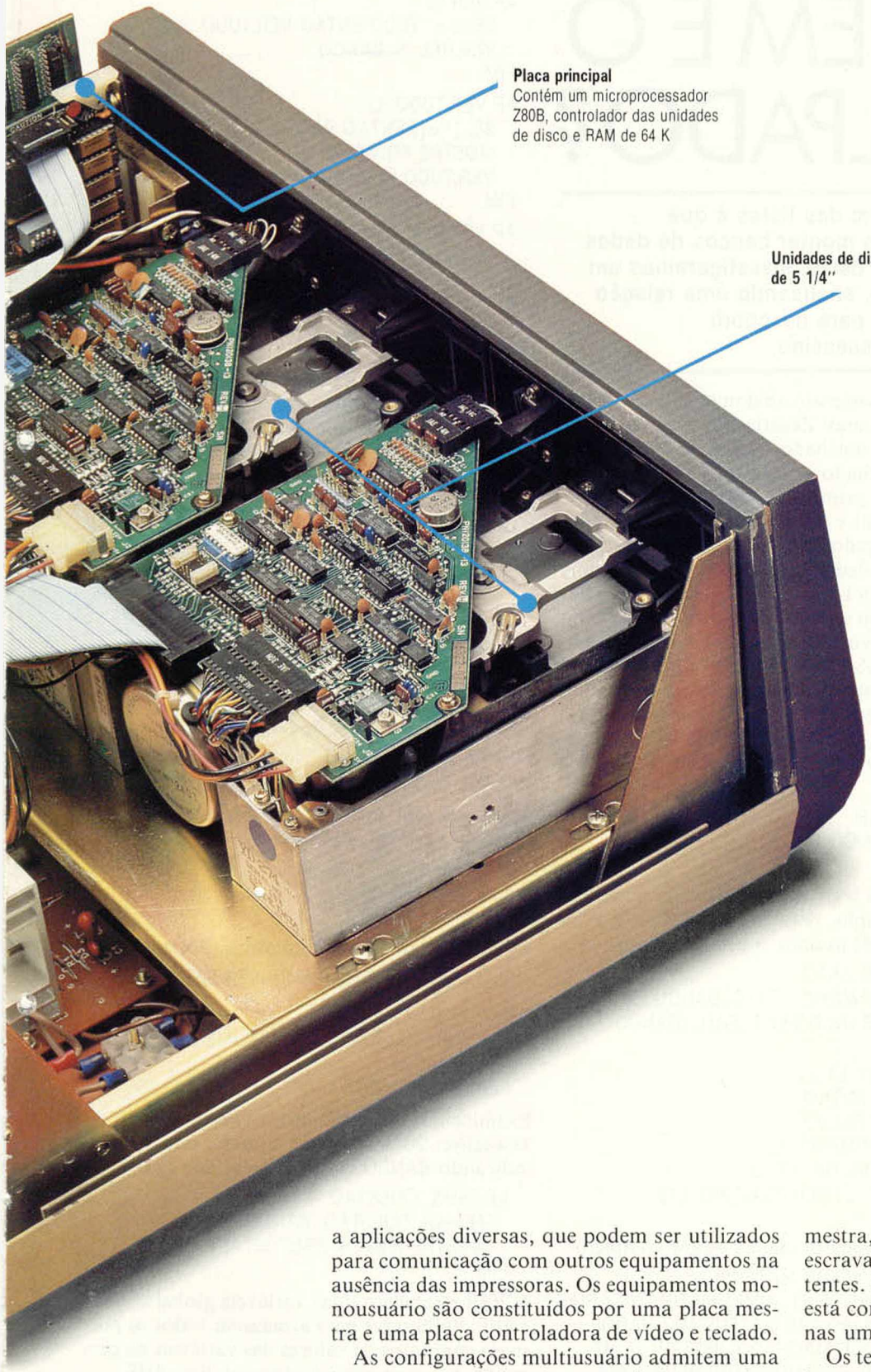
ALFA 3003

De design compacto e funcional, o Alfa 3003 linha Plus, da Dismac, trabalha com até oito módulos de processamento escravo, que garantem desempenho superior ao de qualquer sistema multiusuário.

A linha Plus compreende uma série de equipamentos para aplicações profissionais, incorporando o que há de mais moderno em arquitetura de 8 bits. Esses equipamentos estão dispostos em configurações monousuário e multiusuário (vários terminais, cada um com sua própria unidade de processamento escrava). O sistema aceita até oito módulos de processamento escravo em comunicação com a unidade central de processamento (CPU mestra). Essa unidade, com acesso direto à memória, possui um relógio em tempo real, o que facilita as aplicações utilizadas pelos terminais. Com esses dois recursos, os módulos de processamento escravo oferecem desempenho superior a qualquer outro sistema multiusuário.

As configurações monousuário suportam até quatro discos flexíveis e quatro rígidos, além de uma impressora paralela e duas seriais. Estas últimas ocupam canais de comunicação destinados





DISMAC ALFA 3003 LINHA PLUS

MICROPROCESSADORES

Mestre: Zilog Z80B.
Dedicado: Intel 8085.

CLOCK

6 MHz.

VÍDEO

Monitor de 12", fósforo verde, 24 + 1 linhas de oitenta caracteres.

MEMÓRIA

RAM de 128 K.
ROM de 8 K.

TECLADO

Alfanumérico e numérico reduzido. Tecnologia Reed.

PERIFÉRICOS

Impressoras seriais e paralelas (máximo de seis); unidades de disco flexível, dupla face, dupla densidade (máximo de quatro); unidades de disco rígido (máximo de quatro); interface serial RS232 ou 422; modems Ralf-duplex ou full-duplex; comunicação síncrona ou assíncrona.

LINGUAGENS

COBOL MB, BASIC MB e FORTRAN 80

SISTEMA OPERACIONAL

Sistema operacional multiusuário Dismac Rede versão 1.4, compatível com CP/M e MP/M.

DOCUMENTAÇÃO

Manual de linguagem, manual do sistema operacional e manual do usuário.

a aplicações diversas, que podem ser utilizados para comunicação com outros equipamentos na ausência das impressoras. Os equipamentos monousuário são constituídos por uma placa mestra e uma placa controladora de vídeo e teclado.

As configurações multiusuário admitem uma impressora paralela e cinco canais de comunicação serial, em geral usados para impressoras adicionais, mas que também servem para comunicação com outros equipamentos. Tal como as monousuário, essas configurações também suportam até quatro discos flexíveis e quatro rígidos de capacidades distintas. Possuem uma placa mestra, um controlador de disco rígido e placas escravas em número igual ao dos terminais existentes. No modelo com dois terminais, um deles está conectado à placa mestra, o que exige apenas uma placa escrava.

Os terminais da rede podem ser ligados por cabos co-axiais à unidade central, distante até 1 km. O sistema aceita as linguagens COBOL MB, BASIC MB e FORTRAN 80.

O equipamento permite trabalhar com teleprocessamento e dispõe de protocolos para todas as máquinas de médio e grande portes existentes no mercado.

QUEM É O CULPADO?

Uma vantagem das listas é que nos permitem montar bancos de dados. Com a ajuda delas, investigaremos um terrível crime, analisando uma relação de suspeitos para descobrir quem foi o assassino.

Um terrível assassinato abalou uma pequena aldeia nas montanhas: Zacarias foi atacado e morto a golpes de machado. Sabemos que Mateus e José têm machado, que Jaime e Heitor têm revólver e que a prima Alice tem uma faca. Nas mãos de Mateus e Jaime havia sangue quando foram interrogados pelo delegado da região.

O banco de dados em LOGO sobre esse crime consistirá numa lista de *fatos* — cada um deles compreendendo uma *relação* associada a um ou mais substantivos. No LOGO, representa-se um fato assim: [POSSUI MATEUS MACHADO], ou, em linguagem comum: "Mateus possui um machado". Para dizer que Jaime tinha sangue nas mãos, usamos [ENSANGUENTADO JAIME].

No início da investigação, o banco de dados está vazio:

```
AP PREPARAR
  FACA "BANCO []
FIM
```

Os fatos serão registrados durante a investigação. Por exemplo, para incluir ADICIONE [POSSUI ALICE FACA] usamos o procedimento:

```
AP ADICIONE :FATO
  SE NAO MEMBRO? :FATO :BANCO ENTAO
    FACA "BANCO PENT :FATO :BANCO
FIM
AP MEMBRO? :L1 :L2
  SE :L2 = [] ENTAO
    SAIDA "FALSO
  SE :L1 = PRIMEIRO :L2
    ENTAO SAIDA "VERD
  MEMBRO? :L1 SEMPRIMEIRO :L12
FIM
```

No final, o banco de dados estará completo:

```
[[ENSANGUENTADO MATEUS]
  [ENSANGUENTADO JAIME] [MATOU ZACARIAS
  MACHADO] [POSSUI MATEUS MACHADO]
  [POSSUI JOSE MACHADO] [POSSUI JAIME
  REVOLVER] [POSSUI HEITOR REVOLVER]
  [POSSUI ALICE FACA]]
```

Para imprimir os dados armazenados use VER. Acrescentando "TUDO, serão mostrados todos os dados. No caso de acrescentar o nome de uma relação, somente os fatos associados a ela serão impressos, como em VER "POSSUI.

```
AP VER :S
  SE :S = "TUDO ENTAO VER.TUDO :BANCO
  VER.REL :S :BANCO
FIM
AP VER.TUDO :LI
  SE :LI = [] ENTAO PARE
  MOSTRE PRIMEIRO :LI
  VER.TUDO SEMPRIMEIRO :LI
FIM
AP VER.REL :S :LI
  SE :LI = [] ENTAO PARE
  SE :S = PRIMEIRO PRIMEIRO :LI ENTAO
    MOSTRE PRIMEIRO :LI
  VER.REL :S SEMPRIMEIRO :LI
FIM
```

Agora precisamos criar métodos de consulta ao banco de dados. O tipo mais simples verifica se um fato é considerado verdadeiro. Para isso, usamos o procedimento VERIFIQUE, que informa se um fato está no banco de dados. Por exemplo, VERIFIQUE [POSSUI ALICE FACA] tem como resultado SIM.

```
AP VERIFIQUE :FATO
  SE MEMBRO? :FATO :BANCO ENTAO
    MOSTRE "SIM SENAO MOSTRE "NAO
FIM
```

Seria muito útil, nesta investigação, formular perguntas do tipo "Quem possui um machado?". Para fazer isso empregamos as "variáveis", iniciadas por ?. Podemos então recolocar assim a pergunta:

```
QUAL [POSSUI ?ALGUEM MACHADO]
```

A resposta será uma lista de todos os valores possíveis da variável ?ALGUEM.

```
[?ALGUEM MATEUS]
[?ALGUEM JOSE]
FIM DA RESPOSTA
```

As variáveis podem ser múltiplas. Por exemplo:

```
QUAL [MATOU ?HOMEM ?INSTRUMENTO]
```

resultará na resposta:

```
[?HOMEM ZACARIAS] [?INSTRUMENTO
  MACHADO]
FIM DA RESPOSTA
```

Examinemos os procedimentos que permitem essa análise. QUAL transfere a tarefa para ACHAR, indicando BANCO como a fonte dos fatos.

```
AP QUAL :QUESTAO
  ACHAR :QUESTAO :BANCO
  MOSTRE [FIM DA RESPOSTA]
FIM
```

ACHAR estabelece duas variáveis globais, VARS e ANS. VARS serve para armazenar todos os possíveis conjuntos de valores das variáveis na pergunta, os quais são reunidos na lista ANS.

```
AP ACHAR :QUESTAO :DADOS
  FACA "VARS []
  FACA "ANS []
  COMPARAR :QUESTAO :DADOS
    MOSTREL :ANS
FIM
```

COMPARAR examina cada um dos fatos e, se houver alguma correspondência entre eles, os novos valores em VARS serão acrescentados a ANS. COMPARAR volta a associar VARS à lista vazia, e continua a procurar outras correspondências.

```
AP COMPARAR :QUESTAO :DADOS
  SE :DADOS = [] ENTAO PARE
  SE IGUAL? :QUESTAO PRIMEIRO :DADOS
    ENTAO FACA "ANS PENT :VARS :ANS
  FACA "VARS []
  COMPARAR :QUESTAO SEMPRIMEIRO :DADOS
FIM
```

Suponha que as entradas para IGUAL? sejam [POSSUI ?ALGUEM MACHADO] e [POSSUI JOSE MACHADO]. Nesse caso, IGUAL? tem VERDADEIRO como saída e atribui [?ALGUEM JOSE] a VARS. Se as entradas forem [POSSUI ?ALGUEM MACHADO] e [MATOU ZACARIAS MACHADO], então IGUAL? terá FALSO como saída.

Entretanto, a dificuldade surge quando há mais de uma variável envolvida. VALE? verifica se já foi atribuído algum valor para a variável associada a determinado fato.

Empregaremos uma notação alternativa para comandos condicionais. Se o resultado de TESTE for verdadeiro, as ações indicadas por SEVERD serão executadas; caso contrário, executam-se as ações indicadas por SEFALSO.

```
AP IGUAL? :QUESTAO :FATO
  SE [SETODOS [:QUESTAO = []][:FATO = []]]
    ENTAO SAIDA "VERD
  TESTE PRIMEIRO PRIMEIRO :QUESTAO = "?
  SEVERD SE NAO VALE? PRIMEIRO :QUESTAO
    PRIMEIRO :FATO :VARS ENTAO SAIDA
    "FALSO
  SEFALSO SE NAO [PRIMEIRO :QUESTAO =
    PRIMEIRO :FATO] ENTAO SAIDA "FALSO
  SAIDA IGUAL? SEMPRIMEIRO :QUESTAO
    SEMPRIMEIRO :FATO
FIM
```

Para entender o funcionamento de VALE?, veja o caso em que seus dados são ?INSTRUMENTO, MACHADO e ?HOMEM ZACARIAS. VALE? verifica se INSTRUMENTO pode ter o valor MACHADO. Há três possibilidades: ?INSTRUMENTO possui um valor, que não é MACHADO, fazendo com que a saída de VALE? seja FALSO; ou, então, ?INSTRUMENTO possui o valor MACHADO, e VALE? tem como resultado VERD; ou, ainda, ?INSTRUMENTO não tem um valor, recebendo o valor MACHADO. Essa informação vai para a VARS e a saída é VERD.

```
AP VALE? :NOME :VALORV :LISTAV
  SE :LISTAV = [] ENTAO FACA "VARS
    UENT LISTA :NOME :VALORV :VARS
    SAIDA "VERD
  TESTE :NOME = PRIMEIRO PRIMEIRO
    :LISTAV
  SEVERD SE :VALORV = ULTIMO PRIMEIRO
    :LISTAV ENTAO SAIDA "VERD SENAO
    SAIDA "FALSO
    SAIDA VALE? :NOME :VALOR SEMPRIMEIRO
    :LISTAV
  FIM
```

MOSTREL simplesmente faz com que os itens de ANS sejam impressos um abaixo do outro.

```
AP MOSTREL :LISTAV
  SE :LISTAV = [] ENTAO PARE
  MOSTRE PRIMEIRO :LISTAV
  MOSTREL SEMPRIMEIRO :LISTAV
FIM
```

Perguntas mais complexas

No entanto, a investigação sobre o assassinato dificilmente avançará se não pudermos fazer perguntas mais complexas, do tipo "Que instrumento matou Zacarias e quem possui tal instrumento?". A tradução disso para o LOGO seria:

```
QUAL [MATOU ZACARIAS ?INSTRUMENTO]
[POSSUI ?SUSPEITO ?INSTRUMENTO]
```

Assim, os dados de entrada para QUAL são uma lista de perguntas e os valores encontrados serão aqueles que tornam verdadeiras todas as perguntas. Uma pergunta simples com esse novo tipo de QUAL utiliza a seguinte sintaxe:

```
QUAL [[POSSUI ?UMA FACA]]
```

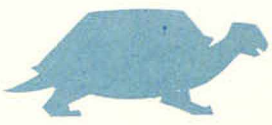
Precisamos alterar um pouco os procedimentos:

```
AP QUAL :QUESTAO
  ACHAR :QUESTAO :BANCO
  MOSTRE [FIM DA RESPOSTA]
FIM
AP ACHAR :QUESTAO :DADOS
  FACA "VARS []
  FACA "ANS []
  COMPARAR :QUESTAO :DADOS
  MOSTREL :ANS
FIM
```

A função de COMPARAR tornou-se agora mais complicada. Tome [[MATOU ZACARIAS ?INSTRUMENTO] [POSSUI ?SUSPEITO ?INSTRUMENTO]] como um exemplo de entrada. COMPARAR examina o banco para encontrar a resposta da primeira pergunta, e acaba fazendo a associação de ?INSTRUMENTO com MACHADO. A rotina passa então para a segunda pergunta ([POSSUI ?SUSPEITO ?INSTRUMENTO]) e repete o processo. Na segunda condição, ela encontra uma correspondência entre o valor de ?INSTRUMENTO — MACHADO e o de ?SUSPEITO — MATEUS. Como não há mais perguntas, essa é uma solução possível.

No entanto, pode haver outros valores que satisfaçam a segunda pergunta, mantendo MACHADO como o valor de ?INSTRUMENTO. Desse modo, COMPARAR retoma o exame dos dados a partir do ponto em que parou, e de fato encontra uma segunda solução com ?SUSPEITO como JOSE. O procedimento não pára aí, mas continua pesquisando em BANCO. Agora, ele chega ao fim sem encontrar outros valores que possam ser associados.

Uma vez que é possível outra solução para a primeira pergunta — que não seja MACHADO —, devemos retornar ao ponto em que foi encon-



Matriz de assassinato

O sr. Pereira foi encontrado morto, com trinta golpes de formão, na carroçaria de um caminhão. A polícia identificou quatro suspeitos: um pedreiro, um açougueiro, uma jardineira e um desenhista. Todos tinham acesso a instrumentos cortantes: a faca do açougueiro, a tesoura da jardineira, o estilete do desenhista e o formão do pedreiro. Um deles foi visto na esquina de uma rua, outro no galpão de um jardim e um terceiro alegou estar dormindo em casa. Aquele que puder ser vinculado ao caminhão é o assassino.
A investigação policial revelou os seguintes fatos:
1) Nenhum deles estava com sua ferramenta de trabalho na noite do crime. **2)** O açougueiro foi localizado no galpão abrindo cartas com um estilete. **3)** Uma testemunha ocular confirmou que o pedreiro se encontrava na esquina de uma rua, onde depois a jardineira acharia sua tesoura. **4)** A jardineira estava em sua cama usando uma faca de açougueiro para preparar um sanduíche de carne.

trada essa correspondência e prosseguir a partir daí. Esse processo é chamado retrocesso. Nesse caso, não há, de fato, outras soluções.

Para não perder o ponto em que se encontra o processo de atribuição de variáveis, COMPARAR coloca os valores atuais em uma pilha — antes de IGUAL? ser usada, já que ela pode alterar as atribuições — para recuperá-los depois. Eis o procedimento completo:

```
AP COMPARAR :QUESTAO :DADOS
  SE QUESTAO = [] ENTAO FACA "ANS
    PENT :VARS :ANS PARE
  SE DADOS = [] ENTAO PARE
  GUARDE :VARS
  TESTE IGUAL? PRIMEIRO :QUESTAO
    PRIMEIRO :DADOS
  SEVERD COMPARAR SEMPRIMEIRO
    :QUESTAO :DADOS
  TRAGA "VARS
  COMPARAR :QUESTAO SEMPRIMEIRO
    :DADOS
FIM
```

Em COMPARAR empregamos uma pilha para manter o controle sobre os valores de VARS, pois, se usássemos uma variável temporária, COMPARAR poderia chamar a si mesma entre o momento de gravar os valores e o de utilizá-los. Sem a pilha, qualquer variável temporária desse tipo poderia ser sobreposta pela chamada seguinte e os valores originais seriam perdidos.

GUARDE coloca um valor no "topo" da pilha, criando antes a variável PILHA, caso ela não exista.

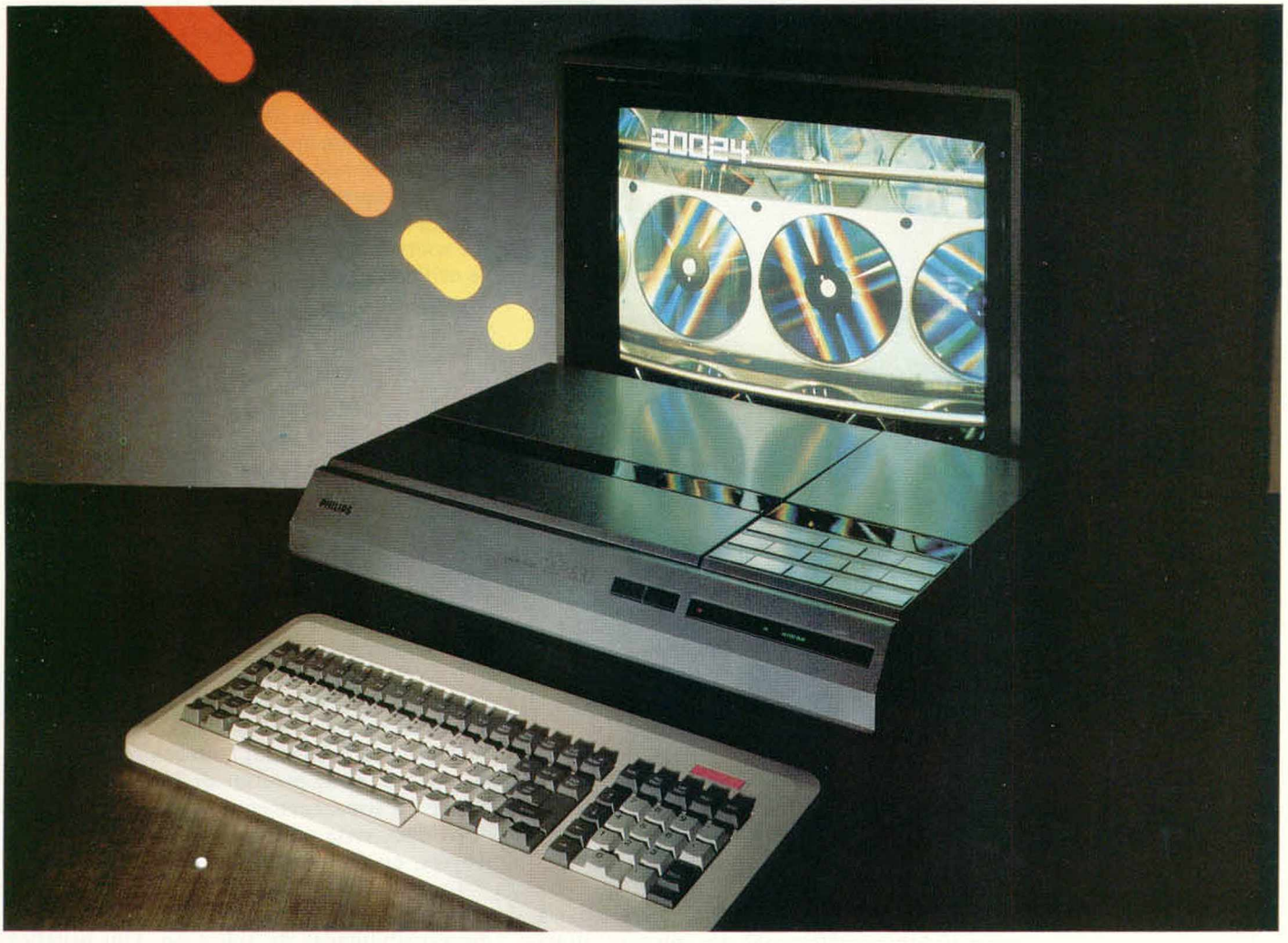
```
AP GUARDE :DADOS
  SE NAO VALOR? :PILHA ENTAO
    FACA "PILHA []
  FACA "PILHA PENT :DADOS :PILHA
FIM
```

TRAGA tira um item da pilha e o atribui como valor de uma variável.

```
AP TRAGA :NOME
  FACA :NOME PRIMEIRO :PILHA
  FACA "PILHA SEMPRIMEIRO :PILHA
FIM
```

São esses os rudimentos de uma linguagem de "programação lógica", em que simplesmente acrescentamos os fatos e as regras a um banco de dados e então o consultamos por meio de descrições lógicas dos dados de que precisamos.

TRILHAS ESPIRAIS



À medida que cresce a produção de discos flexíveis e videodiscos, esses dispositivos se tornam mais acessíveis a usuários domésticos e a pequenas empresas, com toda a sua variedade de aplicações.

Em muitos países, os gravadores de vídeo com discos a laser, considerados artigos de luxo até há poucos anos, já custam menos que os gravadores de videocassete, mesmo proporcionando melhor reprodução de imagem.

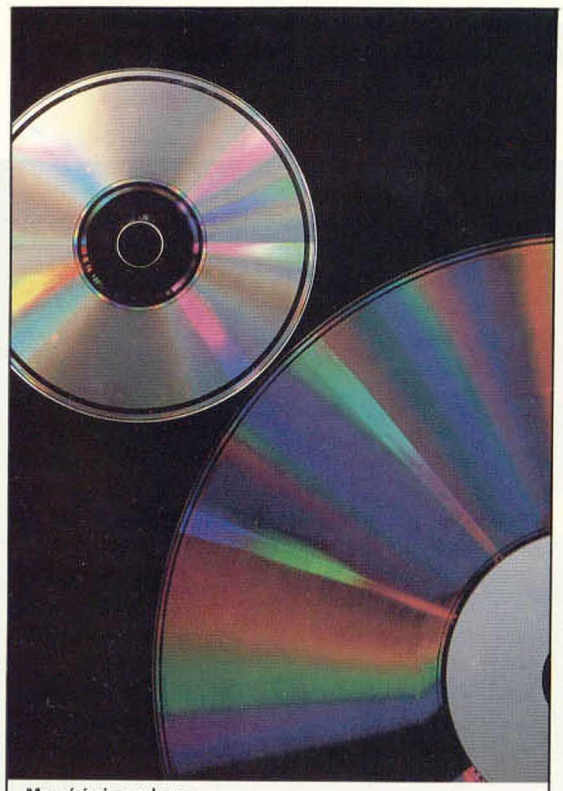
Uma cena normal de televisão é uma imagem dinâmica que pode ser gravada em videoteipe como seqüência ininterrupta. A única maneira de localizar determinada parte da seqüência gravada é movendo a fita para a frente, em velocidade normal ou acelerada, ou usando a tecla de retrocesso.

Os discos armazenam as imagens em quadros separados, sendo possível ir diretamente e com precisão a qualquer um deles. A localização de uma imagem no disco pode ser descrita em termos de trilha e setor, e um microprocessador mantém um catálogo atualizado do endereço de cada uma. O microprocessador supervisiona o acesso ao disco e permite congelar a imagem ou passá-la em câmara lenta e som estereofônico.

Tarefas tais como girar um prato giratório em velocidade alta e constante e posicionar uma cabeça de leitura exatamente em determinado lugar da superfície não são as realizações mais surpreendentes da tecnologia do disco. O maior desafio consistiu em obter a altíssima densidade de armazenamento dos discos, com o tamanho de um LP de 12 polegadas.

Se você já usou gráficos de alta resolução em seu micro, sabe que a imagem da televisão é composta de pontos (pixels) de luz isolados — quanto mais pontos houver na tela, melhor a imagem apresentada — e que o armazenamento de telas de alta resolução necessita de grande quantidade de memória.

O laser chega ao lar
Os toca-discos a laser tendem a ser tão baratos quanto os microcomputadores, e não será raro encontrar os dois interligados em casa. Adquirindo os discos e o software apropriados, você poderá ter acesso a bancos de dados pictóricos e a sofisticados programas de treinamento, bem como a jogos de aventura.



Memória inovadora
Discos a laser de 12 polegadas e compactos menores servem para armazenar informações de vídeo e áudio, além dos dados digitais hoje gravados em discos flexíveis e fitas cassete. A desvantagem é que não se pode gravar nesses discos e tecnicamente eles são apenas ROM. Assim, você não grava seus próprios programas e informações em discos a laser; deve utilizar software pré-gravado.

Se uma tela de televisão tiver resolução de 640 x 256 pixels (equivalente à dos melhores micros), então o armazenamento de um segundo de vídeo (a 25 quadros por segundo) exigirá 25 x 20 Kbytes de memória.

Um minuto de um programa de televisão ocuparia 30 Mbytes (30 megabytes, ou seja, 30 milhões de bytes) e um episódio de um seriado preferido chegaria a ocupar 1 Gbyte (1 gigabyte, ou seja, 1.000 megabytes) de memória.

Se você utilizasse discos flexíveis de densidade simples e face simples, usados na maioria dos microcomputadores, seriam necessários mais de 6.000 discos e uma semana de trabalho!

A resposta a esse desafio consiste em gravar os dados de maneira diminuta. Um gravador de disco a laser grava os dados com um raio laser da espessura de um fio de cabelo sobre uma placa metálica revestida com uma camada rígida translúcida. Um raio de baixa potência é usado para ler os dados no disco.

Os lasers são usados como uma agulha de leitura-gravação para esse tipo de disco porque são dispositivos de baixa tolerância e fina resolução. Nenhuma outra técnica conhecida pode ler e gravar com eficiência tantos dados em tão pouco espaço.

O formato é uma combinação das técnicas usadas em discos musicais e disquetes flexíveis. Os sulcos de um disco de música formam uma espiral contínua e as paredes dos sulcos contêm uma representação gravada das formas de onda do som.

A maioria dos usuários de micro sabe que as trilhas do disco flexível são círculos concêntricos e que a informação é gravada magneticamente sobre o disco e armazenada de modo digital em padrões de 0 e 1.

O formato do disco laser usa trilhas e não sulcos, mas em forma de espiral. A informação é gravada opticamente sobre o disco em padrões de 0 e 1, mas não pode ser apagada. Os dígitos 0 e 1 (que representam os padrões de pontos que compõem uma imagem de televisão) são gravados sobre a superfície do disco pelos minúsculos orifícios que o laser faz queimando a película de metal para representar os dígitos 1 e deixando-a intata para representar os 0. Cada orifício tem 0,5 μm (meio micrômetro, ou seja, 0,0005 mm) de largura e 0,1 μm (um décimo de micrômetro, ou seja, 0,0001 mm) de profundidade. Portanto, 1 cm^2 do disco pode conter 400 milhões desses orifícios.

Essa miniaturização extraordinária é suficiente para suprir a demanda de armazenamento de vídeo. Um disco de 35 cm de diâmetro contém 54.000 quadros de televisão por lado, ou aproximadamente 36 minutos de gravação. Os cálculos anteriores referiam-se à resolução de gráficos de computador em preto e branco, enquanto o disco deve armazenar uma trilha de áudio e também informações sobre as cores de cada ponto da imagem da televisão.

Um quadro colorido, com sua respectiva trilha sonora, pode precisar de 100 Kbytes de espaço para armazenamento. Para 54.000 quadros desse tipo, seria necessário usar 5,4 milhões de Kbytes, ou 5,4 Gbytes.

Tendo resolvido o problema de limitação de armazenamento, os discos laser oferecem grandes possibilidades de aplicação. Um benefício importante consiste em remover um dos maiores obstáculos em processamento de dados — coleta e entrada de informação. Mesmo assim, é necessário sentar diante de um terminal e digitar representações codificadas da informação para o sistema.

Se, ao invés disso, você puder apontar uma câmara para os dados e deixar que ela armazene a informação visual no disco enquanto você digita apenas os detalhes dos índices das imagens gravadas, a carga de trabalho apresentará notável diminuição.

Os toca-discos a laser variam muito no que se refere à sofisticação. Já se encontram disponíveis gravadores de vídeo com discos a laser para uso doméstico. Eles passam imagens como um gravador comum de videocassete, mas com a vantagem adicional de oferecer alta definição tanto em velocidade normal quanto em câmara lenta. Você também pode escolher quadros isolados em alguns equipamentos simplesmente digitando seu número num controle manual. Os que fazem isso — fabricados por empresas co-

mo a Pioneer e a Philips — ainda são caros e de uso exclusivamente profissional.

O sistema mais simples consiste em usar uma interface IEEE ou RS232 para que o computador selecione um determinado quadro pelo seu número. Um software apropriado mantém uma lista dos quadros disponíveis num banco de dados e os seleciona quando necessário.

A Philips levou a idéia um passo adiante e incorporou um microcomputador simples a seus modelos mais recentes. Assim, pode-se carregar um programa para determinado disco a partir de um cartucho EPROM conectado à máquina ou então a partir do próprio disco a laser. Cada disco a laser armazena duas trilhas de áudio e uma de vídeo. Isso permite que um único disco tenha trilha sonora em duas línguas, por exemplo. Contudo, se a segunda trilha de áudio não for necessária, poderá ser usada para armazenar um programa de computador.

Assim, temos um banco de 54.000 imagens de qualidade sob controle do computador. O estágio final seria combinar as imagens do disco a laser e o texto do computador. Isso poderia ser feito com dois monitores separados, ou combinando as duas entradas de vídeo, ou ainda usando-se um monitor com seu próprio gerador de teletexto.

Surge assim um meio de comunicação inteiramente novo — o vídeo interativo. O usuário e o software podem dirigir o que aparece na tela, tanto as cenas de ação quanto as imagens congeladas, lendo o disco em qualquer ordem.

Sua aplicação mais óbvia é para um banco de dados pictórico. O usuário pode fazer perguntas ao computador, que retira as informações relevantes de um banco de dados e instrui o equipamento a apresentar o quadro de vídeo apropriado. As vantagens para efetuar consultas em bibliotecas e escolas são evidentes.

O vídeo interativo já avançou um estágio, envolvendo o usuário nas seqüências mostradas na tela: um programa de treinamento explica alguma coisa com um pequeno trecho de um filme e então faz perguntas, recapitulando se houver necessidade, ou aprofundando-se nos detalhes.

O usuário pode até modificar o roteiro e o final de um filme pronto, "dirigindo-o", ou ampliar as possibilidades dos jogos de aventura.

Além do problema do custo do equipamento e da fabricação dos discos a laser, novos recursos precisam ser desenvolvidos no planejamento e na produção de discos interativos, tanto em termos de software de computador quanto na execução do roteiro e da filmagem das cenas.

Muitas empresas pequenas em todo o mundo estão começando a enfrentar esses desafios e uma multinacional, a Computer Assisted Televideo (CAT), já oferecia, em 1985, um serviço completo para clientes — escolha e instalação do equipamento, planejamento e produção de discos e criação de software adequado. Suas atividades, porém, se limitaram à produção de programas de treinamento.

No entanto, com o tempo, os discos a laser poderiam ser vendidos, com software, por preço um pouco superior ao dos filmes e videoteipes. Considerando a qualidade dos programas de entretenimento e educativos por eles possibilitados, o mercado, sem dúvida, os absorveria.

Vídeo interativo
Muitos toca-discos a laser possuem uma interface IEEE ou RS232, que lhes permite ser controlados por microcomputadores.
Um conjunto típico seria um computador com um banco de dados armazenado em discos flexíveis e relacionado com as imagens no disco a laser.
O usuário escolhe um item no banco de dados e, a seguir, o computador faz com que o toca-discos a laser selecione e exiba a imagem apropriada, tendo como referência o número do quadro. O sistema aqui mostrado adota o recurso de combinar a saída do computador com as imagens do disco a laser na mesma tela.

SIGA O MAPA

Muito úteis na simplificação dos circuitos lógicos, os mapas de Karnaugh não substituem todas as simplificações algébricas, mas reduzem o esforço para fatorar expressões complexas.

Os mapas de Karnaugh (também conhecidos como "mapas-k") são na verdade extensões dos diagramas de Venn, já examinados anteriormente.

Esses diagramas possibilitam representar expressões lógicas por meio de figuras. Já um mapa-k assume formas diversas, dependendo do número de letras (ou variáveis) diferentes que houver na expressão a se simplificar.

O mapa-k é mais útil para expressões que contêm duas, três ou quatro variáveis.

Duas variáveis. Cada quadrado de um mapa-k de duas variáveis (há $2^2 = 4$ quadrados) representa uma função E, como se mostra abaixo:

	A	$\bar{A}$
B	A.B	$\bar{A}$.B
$\bar{B}$	A.$\bar{B}$	$\bar{A}$.$\bar{B}$

Representa-se a expressão $AB + A\bar{B}$ como um mapa-k, colocando os 1 nos quadrados próprios:

	A	$\bar{A}$
B	1	0
$\bar{B}$	1	0

Aqui estão três outros exemplos que representam as expressões $\bar{A}\bar{B}$, $AB + \bar{A}\bar{B}$ e $AB + \bar{A}B + A\bar{B}$, respectivamente:

	A	$\bar{A}$
B	0	0
$\bar{B}$	0	1

	A	$\bar{A}$
B	1	0
$\bar{B}$	0	1

	A	$\bar{A}$
B	1	1
$\bar{B}$	1	0

Três variáveis. Nesse caso, o número de quadrados aumenta em um fator de dois ($2^3 = 8$ quadrados). O mapa-k básico é:

	A	$\bar{A}$	
B	A.B.C	$\bar{A}$.B.C	C
	A.$\bar{B}$.C	$\bar{A}$.$\bar{B}$.C	
$\bar{B}$	A.$\bar{B}$.$\bar{C}$	$\bar{A}$.$\bar{B}$.$\bar{C}$	$\bar{C}$
B	A.B.$\bar{C}$	$\bar{A}$.B.$\bar{C}$	

Eis duas expressões, $AC + \bar{A}B\bar{C}$ e $AB + \bar{A}C$, representadas como mapas-k:

	A	$\bar{A}$	
B	1	0	C
	1	0	
$\bar{B}$	0	0	$\bar{C}$
B	0	1	

	A	$\bar{A}$	
B	1	1	C
	0	1	
$\bar{B}$	0	0	$\bar{C}$
B	1	0	

$$ABC + A\bar{B}C + \bar{A}B\bar{C} =$$
$$= AC(B + \bar{B}) + \bar{A}B\bar{C}$$
$$= AC + \bar{A}B\bar{C}$$

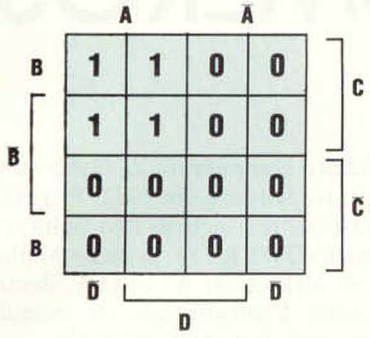
$$ABC + AB\bar{C} + \bar{A}BC + \bar{A}\bar{B}C =$$
$$= AB(C + \bar{C}) + \bar{A}C(B + \bar{B})$$
$$= AB + \bar{A}C$$

Ambas as expressões foram simplificadas usando-se a lei booleana que estabelece que um conjunto A operado em OU com sua negativa ($\bar{A}$) produz 1 — o conjunto Universo ou Identidade.

Quatro variáveis. Os mapas se tornam mais complexos (eles têm $2^4 = 16$ quadrados), mas mesmo assim são bastante simples de interpretar, de acordo com o diagrama básico:

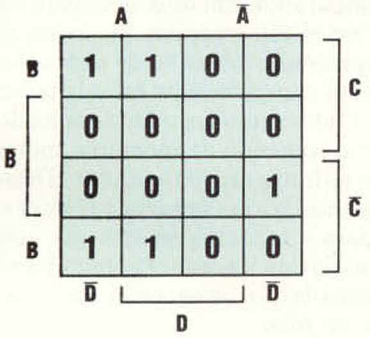
	A		$\bar{A}$		
B	A.B.C.$\bar{D}$	A.B.C.D	$\bar{A}$.B.C.D	$\bar{A}$.B.C.$\bar{D}$	C
	A.$\bar{B}$.C.$\bar{D}$	A.$\bar{B}$.C.D	$\bar{A}$.$\bar{B}$.C.D	$\bar{A}$.$\bar{B}$.C.$\bar{D}$	
$\bar{B}$	A.$\bar{B}$.$\bar{C}$.$\bar{D}$	A.$\bar{B}$.$\bar{C}$.D	$\bar{A}$.$\bar{B}$.$\bar{C}$.D	$\bar{A}$.$\bar{B}$.$\bar{C}$.$\bar{D}$	$\bar{C}$
B	A.B.$\bar{C}$.$\bar{D}$	A.B.$\bar{C}$.D	$\bar{A}$.B.$\bar{C}$.D	$\bar{A}$.B.$\bar{C}$.$\bar{D}$	
	$\bar{D}$			$\bar{D}$	
		D			

Observe agora um mapa-k com simplificação correspondente:



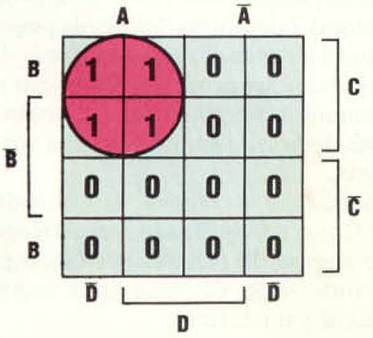
$$ABC\overline{D} + ABCD + A\overline{B}C\overline{D} + A\overline{B}CD =$$
$$= ABC(\overline{D} + D) + A\overline{B}C(\overline{D} + D)$$
$$= ABC + A\overline{B}C$$
$$= AC(B + \overline{B})$$
$$= AC$$

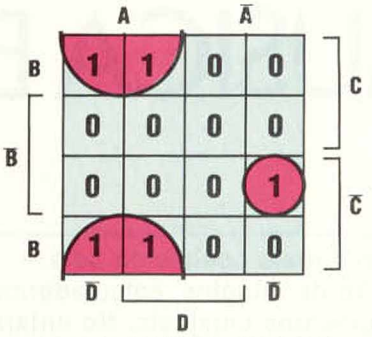
Eis aqui outro exemplo:



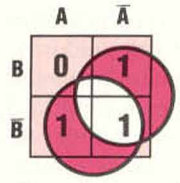
$$ABC\overline{D} + ABCD + \overline{A}BC\overline{D} + AB\overline{C}\overline{D} + AB\overline{C}D$$
$$= ABC(\overline{D} + D) + \overline{A}BC\overline{D} + AB\overline{C}(\overline{D} + D)$$
$$= ABC + \overline{A}BC\overline{D} + AB\overline{C}$$
$$= AB(C + \overline{C}) + \overline{A}BC\overline{D}$$
$$= AB + \overline{A}BC\overline{D}$$

O exame atento da disposição dos 1 nos dois mapas-k anteriores revela certos padrões. No primeiro exemplo, todos os quadrados com AC em suas expressões estão preenchidos com 1. No segundo exemplo, isso acontece com todos os quadrados com AB. Daí, um método mais fácil de simplificar expressões booleanas consiste em inspecionar um mapa-k. Considere:





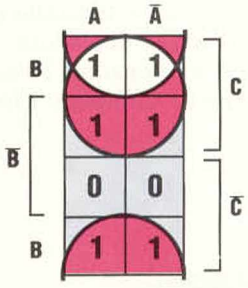
Com um pouco de prática, podem-se escolher grupos de dois, quatro ou oito 1 para formar termos mais simples. Por exemplo, examinemos a expressão: $\overline{A}B + A\overline{B} + \overline{A}\overline{B}$.



Usando um mapa-k de duas variáveis, podemos escolher dois grupos de 1. O primeiro representa todas as ocorrências de NÃO(B) e o outro todas as ocorrências de NÃO(A), de modo que a expressão é simplificada para $\overline{A} + \overline{B}$, ou, lembrando a lei de Morgan, para: $\overline{A.B}$. É possível chegar a essa conclusão mais diretamente examinando o mapa-k?

Um exemplo mais difícil envolve uma expressão de três variáveis:

$$ABC + A\overline{B}C + \overline{A}BC + AB\overline{C} + \overline{A}B\overline{C} + \overline{A}\overline{B}C$$



O grupo de quatro 1 na parte de cima do mapa-k representa todos os casos possíveis em que C é verdadeira. A primeira e última carreiras do mapa representam todos os casos possíveis em que B é verdadeira. Assim, a expressão simplificada será: $B + C$.

No próximo artigo da série trataremos do uso dos mapas de Karnaugh na simplificação de expressões booleanas com quatro variáveis. Então, ao mostrar como os mapas-k são usados no processo de desenhar circuitos, reuniremos todos os aspectos até agora apresentados.

EXERCÍCIO 4

Desenhe mapas-k com três variáveis para simplificar as seguintes expressões booleanas:

a) $\overline{A}.B.C + \overline{A}.\overline{B}.C + \overline{B}.\overline{C} + \overline{A}.B.\overline{C}$

b) $A.\overline{B}.\overline{C} + \overline{A}.B.\overline{C} + A.B.\overline{C}$

Resposta na próxima matéria desta seção.

MÚSICA E NÚMEROS

A Casio é mais conhecida pela produção de relógios, calculadoras e instrumentos musicais. No entanto, tem lançado vários computadores portáteis e de bolso para penetrar no mercado da informática.

Apesar de responsável por cerca de 50% do mercado mundial de calculadoras, no início da década de 80 a Casio tinha apenas 3.300 funcionários, espalhados pelo mundo. Em 1983, seu faturamento foi de 29 milhões de dólares, quantia irrisória para um grande produtor de equipamentos eletrônicos. Tony Manton, diretor de vendas de calculadoras da empresa, declarou que essa é uma atitude deliberada: "Somos uma companhia pequena e conservadora. Grandes campanhas publicitárias não fazem parte de nosso estilo".

A empresa foi fundada no Japão por cinço irmãos da família Kashio, após a Segunda Guerra Mundial (1939-45). Naquela época, era conhecida como Kashio Seisakujo e fabricava equipamentos de escritório. No início da década de 50, produziu a calculadora 14-A Relay, que pesava 130 kg e tinha o tamanho de uma mesa de escritório. Outras calculadoras foram fabricadas e em 1957 a companhia passou a chamar-se Casio Computer Company Ltd.

O Centro de Pesquisa e Desenvolvimento, em Tóquio.

Liderança mundial
Tadao Kashio, presidente da Casio, é um dos cinco membros da família Kashio que estão no topo da estrutura da empresa.

O primeiro computador de bolso da Casio, o FX-702P, foi lançado em 1982. Projetado para uso científico, seu teclado não tinha a disposição comum (QWERTY): as teclas alinhavam-se em ordem alfabética. As vendas, desanimadoras, levaram à substituição da máquina pela FX-700P, com teclado QWERTY.

Seguiu-se o PB-100, computador de bolso para uso comercial. Seu design assemelhava-se ao do FX-702P. Possuía, ainda, um visor de cristal líquido (LCD) e teclas numéricas.

A Casio produziu, além disso, gravadores cassete, impressoras/plotters e cartões de RAM especiais para as duas máquinas.

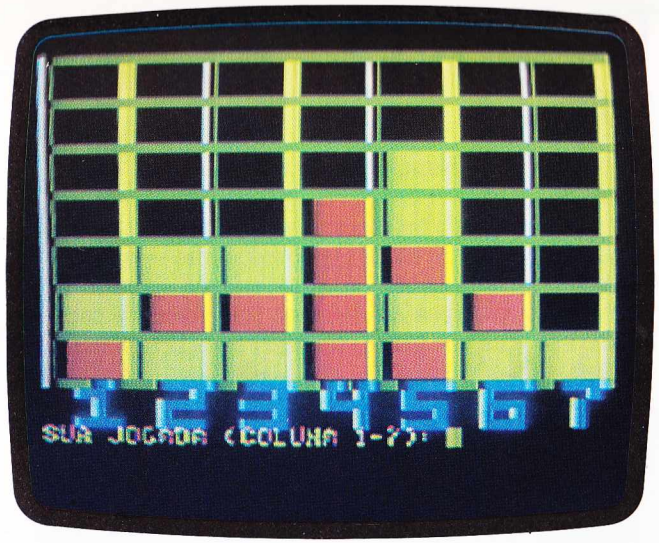
Um substituto recente do PB-100 é o FX-750P. Esse computador tem dois slots para cartões de RAM. Neles são inseridas pequenas placas de metal (que têm o tamanho de uma caixa de fósforos) com capacidade de armazenamento de 4 Kbytes. Cada cartão possui uma pilha de 3 V que mantém o conteúdo da memória, mesmo quando é ele retirado do computador. Desse modo, os programas não se perdem e podem ser carregados para a máquina sempre que requeridos. As pilhas duram um ano. Quando há necessidade de trocá-las, os programas são recarregados a partir de fitas.

A Casio também produz o FP-200, computador portátil equipado com 8 Kbytes de RAM (capacidade que se amplia para 32 Kbytes). O FP-200 tem um LCD com 8 linhas de vinte caracteres e capacidade de resolução gráfica de 160 x 64 pixels.

Posteriormente, a Casio lançou o SL-800. Trata-se de uma calculadora de baixo custo, com peso e tamanho aproximados de um cartão de crédito. Essa forma delgada resulta de um processo de fabricação conhecido como "filming", no qual os componentes são impressos em filme laminado, em vez de serem soldados em placas de circuitos convencionais.

Em muitos países europeus e do Extremo Oriente, a Casio distribui computadores comerciais e microcomputadores padrão MSX. Em outros, a companhia concentrou-se na produção de calculadoras e de computadores de pequeno porte, visando à expansão regrada. Uma das metas da empresa, nesse processo, é mostrar ao público consumidor em potencial que computadores portáteis de bolso são mais do que simples calculadoras.

Com relação a futuros lançamentos nessa área, a Casio tem utilizado a experiência adquirida no campo do teclado eletrônico para produzir uma série de máquinas baseadas na interface digital MIDI.

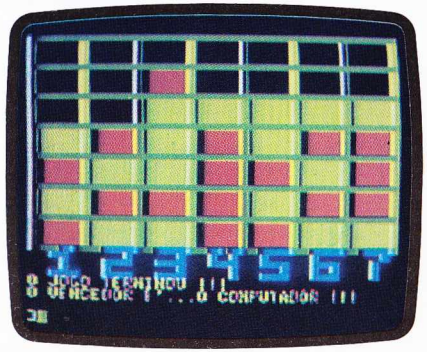


O segredo do jogo

O programa divide a tela, formando um tabuleiro de 49 casas. Seu objetivo é ocupar cinco casas consecutivas — horizontais ou verticais — antes que o computador o faça.

O segredo do jogo está em atrapalhar a jogada do adversário e, ao mesmo tempo, procurar casas estratégicas que permitam conectar cinco. Em último caso, tente o empate.

```
1050  PRINT "APOS A SUA JOGADA, E
      U TEREI MINHA CHANCE. O PRIM
      EIRO QUE CONECTAR 5 SERA O V
      ENCEDOR."
1060  VTAB 22: INPUT "TECLE <RETU
      RN> PARA CONTINUAR:";RES$
1990  RETURN
2000  REM  *** INICIO ***
2010  GR
2011  COLOR= 0
2012  FOR I = 0 TO 35
2013  HLIN 0,35 AT I
2014  NEXT
2015  COLOR= 15
2016  FOR I = 0 TO 35 STEP 5
2017  HLIN 0,35 AT I: VLIN 0,35 AT
      I
2018  NEXT I
2020  COLOR= 6
2021  VLIN 35,39 AT 3
2022  HLIN 2,4 AT 39
2023  PLOT 2,36
2024  HLIN 7,9 AT 35: HLIN 7,9 AT
      37: HLIN 7,9 AT 39
2025  PLOT 9,36: PLOT 7,38
2030  HLIN 12,14 AT 35: HLIN 12,1
      4 AT 37: HLIN 12,14 AT 39: PLOT
      14,36: PLOT 14,38: HLIN 17,1
      9 AT 37: VLIN 35,39 AT 19: VLIN
      35,36 AT 17
2040  HLIN 22,24 AT 35: HLIN 22,2
      4 AT 37: HLIN 22,24 AT 39: PLOT
      22,36: PLOT 24,38: HLIN 27,2
      9 AT 35: HLIN 27,29 AT 37: HLIN
      27,29 AT 39
2050  PLOT 27,38: PLOT 29,38: PLOT
      27,36: HLIN 32,34 AT 35: PLOT
      34,36: VLIN 37,39 AT 33
2900  DEF  FN C(X) = (X - 1) * 5 +
      1
2910  DEF  FN P(X) = (X - 1) * 5 +
      9
2990  RETURN
3000  REM  ************************
      *
3001  REM  ************************
      *
```



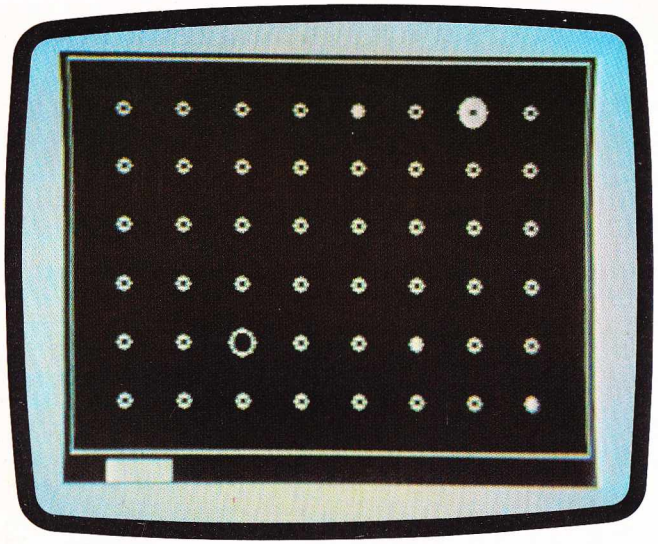

```
3010  HOME :PL = 1: INPUT "SUA JO
      GADA (COLUNA 1-7): ";RES$
3020 RES =  VAL (RES$)
3021  IF RES < 1 OR RES > 7 OR RE
      S <  > INT (RES) THEN  HOME
      : PRINT "SELECIONE UM NUMERO
       ENTRE 1 E 7 :": FOR A = 1 TO
      1000: NEXT A: GOTO 3010
3030 AX =  FN C(RES)
3040  IF  SCRN( AX,1) <  > 0 THEN
      HOME : PRINT "ESTA COLUNA E
      STA' CHEIA... ": FOR A = 1 TO
      1000: NEXT A: GOTO 3010
3050  FOR J = 1 TO 7: COLOR=  FN
      P(PL):JX =  FN C(J): FOR K =
      JX TO JX + 3: HLIN AX,AX + 3
       AT K: NEXT K
3055  IF J = 7 THEN 3080
3060 LX =  FN C(J + 1): IF  SCRN(
      AX,LX) <  > 0 THEN J = 7: GOTO
      3080
3065  COLOR= 0
3070  FOR K = JX TO JX + 3: HLIN
      AX,AX + 3 AT K: NEXT K
3080  NEXT J
3090  FOR I = 1 TO 7: FOR J = 1 TO
      3: FOR K = J TO J + 4:IX =  FN
      C(I):KX =  FN C(K)
```

```
3100  IF  SCRN( IX,KX) <  > FN P
      (PL) THEN K = J + 4: NEXT K:
       GOTO 3110
3105  NEXT K: RETURN
3110  NEXT J,I
3120  FOR J = 1 TO 7: FOR I = 1 TO
      3: FOR K = I TO I + 4:KX =  FN
      C(K):JX =  FN C(J)
3130  IF  SCRN( KX,JX) <  > FN P
      (PL) THEN K = I + 4: NEXT K:
       GOTO 3140
3135  NEXT K: RETURN
3140  NEXT I,J
3150  IF PL = 2 THEN PL = 1: GOTO
      3010
3160 PL = 2:RES =  INT ( RND (1) *
      7) + 1:AX =  FN C(RES): IF  SCRN(
      AX,1) <  > 0 THEN 3160
3170  GOTO 3050
4000  REM  ************************
      *******
4001  REM  ******  TUDO FEITO ***
      *******
4002  REM  ************************
      *******
4010  HOME : PRINT : PRINT "O JOG
      O TERMINOU !!!": PRINT "O VE
      NCEDOR E'...";
4020  IF PL = 1 THEN  PRINT "VOCE
       !!!": RETURN
4030  PRINT "O COMPUTADOR !!!": RETURN
```

PACMAN
PARA CP400

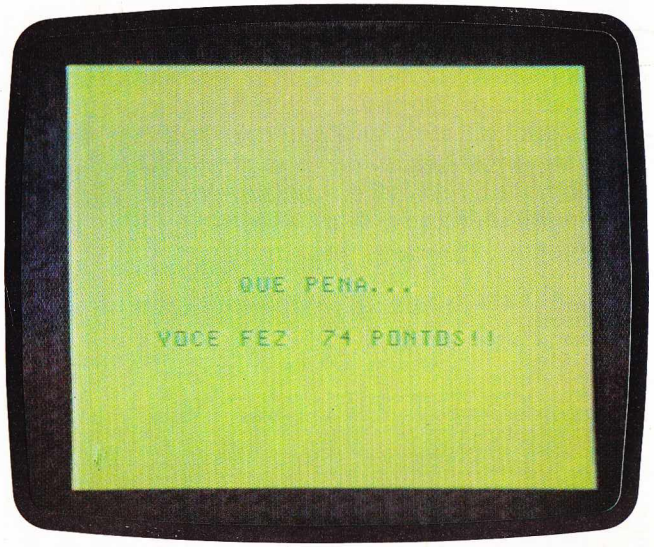
Este jogo utiliza os recursos de alta resolução do CP-400. O jogo começa com a tela repleta de pequenos círculos, alguns dos quais preenchidos por cor. Trata-se de cargas especiais, que dão pontos e combustível extra ao jogador. Este é representado por um círculo maior, vazado, enquanto o inimigo corresponde a um círculo, de mesmo tamanho, cheio. Você deve evitar o inimigo e para tanto pode sair da tela por um lado e voltar pelo outro. Seu objetivo é passar pelas cargas antes do inimigo, do contrário elas perdem seu valor extra. Na parte inferior da tela encontra-se o marcador de combustível. O controle dos movimentos é feito pelas setas do teclado.

```
100 REM ********************************
110 REM *                              *
120 REM *  PARA CP-400 COM 16 OU 64K   *
130 REM *          P A C M A N         *
140 REM *                              *
150 REM ********************************
160 REM *  ESTABELECE ALTA RESOLUCAO   *
170 CLS:PMODE4,1:PCLS:SCREEN1,1
180 CLEAR:DIM P(20,20),K$(4),B(20,20)
190 K$(1)=CHR$(9):K$(2)=CHR$(94):K$(3)=
    CHR$(8):K$(4)=CHR$(10)
200 X=8:Y=6:F=50
210 PCLS:IX=INT(256/(X+1)):IY=
    INT(182/(Y+1))
220 REM *   DESENHA A TELA DO JOGO     *
230 LINE(3,3)-(252,178),PSET,B
240 LINE(20,182)-(20,190),PSET:LINE-
    (F,190),PSET:LINE-(F,182),PSET:
    LINE-(20,182),PSET:PAINT(21,185),1
250 DZ=2+(X+Y)/10
260 N=X*Y
270 REM * POE AS CARGAS NA NOVA TELA   *
280 FOR I=1 TO X:FOR J=1 TO Y
290 CIRCLE(IX*I,IY*J),3:B(I,J)=0:
    IF RND(X+Y)=1 THEN P(I,J)=RND(23):
    PAINT(IX*I,IY*J),1:SOUND P(I,J)*10,1
    ELSE P(I,J)=1:PLAY "L220;A"
300 NEXT J,I
310 DX=0:DY=0
320 XO=RND(X):YO=RND(Y):X1=RND(X):
    Y1=RND(Y)
330 IF ABS(XO-X1)<4 OR ABS(YO-Y1)<4
    THEN 320
340 REM * DESENHA JOGADOR E INIMIGO   *
350 CIRCLE(XO*IX,YO*IY),6:
    CIRCLE(X1*IX,Y1*IY),7:
    PAINT(X1*IX+4,Y1*IY),1:
    PAINT(X1*IX+4,Y1*IY),0
360 REM *    MANUTENCAO DAS CARGAS     *
370 IF P(XO,YO)=0 THEN 390
380 FOR I=1 TO P(XO,YO): PLAY "L170;"
    +STR$(INT(I/2+1)):F=F+1:LINE(F,182)
    -(F,190),PSET:NEXT I:T=T+P(XO,YO):
    N=N-1:IF N=0 THEN X=X+1:Y=Y+1:GOTO
    210
390 A=XO*IX:B=YO*IY
400 IF P(XO,YO)<>0 THEN PAINT (A,B),1:
    PAINT (A,B),0
410 P(XO,YO)=0: GOSUB 420: GOTO 430
420 LINE(F,182)-(F,190),PRESET: F=F-1:
    PLAY"L150;C":IF F=19 THEN PRINT@
    168, "SEM COMBUSTIVEL":GOTO 690
    ELSE RETURN
430 REM *     VERIFICA O TECLADO       *
440 A$=INKEY$: IF A$=" " THEN B(XO,YO)=
    B(XO,YO)+3: PLAY "V30;L200;O1;G":
    PSET(A+RND(5)-3,B+RND(5)-3):
    GOSUB 420: GOTO 440 ELSE IFA$=""
    THEN 490
450 IF A$=K$(1) THEN DX=1:DY=0:GOTO 490
460 IF A$=K$(2) THEN DY=-1:DX=0:GOTO 490
470 IF A$=K$(3) THEN DX=-1:DY=0:GOTO 490
```



Fora da tela
Você é o círculo maior, vazado, e seu inimigo está na parte superior da tela. Os pequenos círculos são cargas especiais, que você precisa atingir antes dele. Use seus recursos: o inimigo só anda dentro da tela, enquanto você pode pular fora.

Combustível extra
Fique de olho no combustível. Para aumentar suas reservas, procure as cargas especiais. Ao final, o computador apresentará o total de pontos.

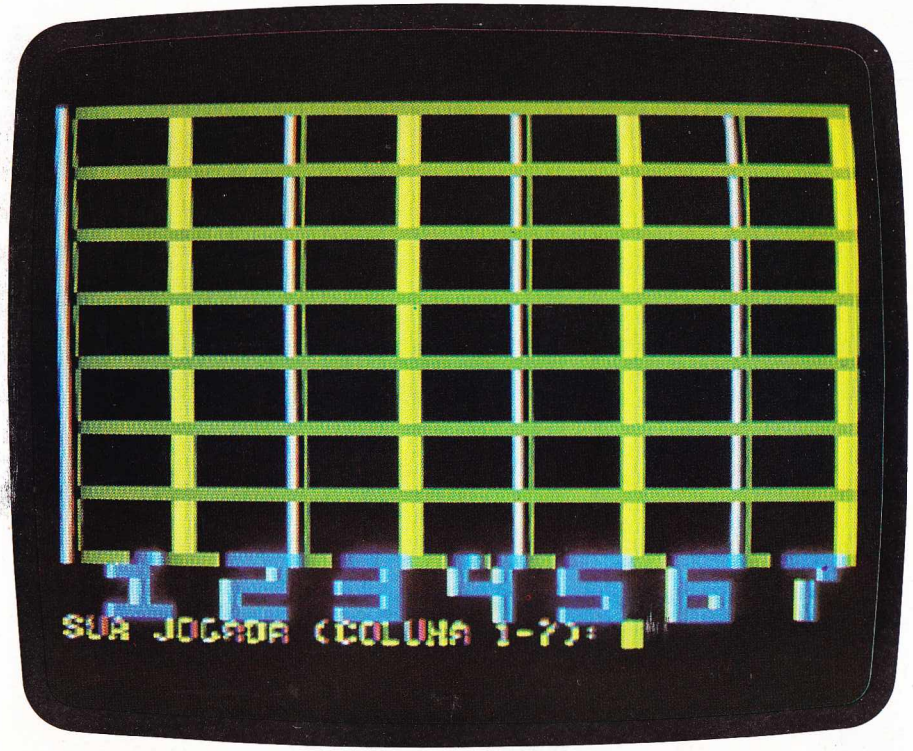



```
480 IF A$=K$(4) THEN DY=1:DX=0
490 XO=XO+DX: IF XO=0 THEN XO=X
500 IF XO>X THEN XO=1
510 YO=YO+DY: IF YO=0 THEN YO=Y
520 IF YO>Y THEN YO=1
530 REM *   DESENHA NOVAS POSICOES
540 CIRCLE(A,B),6,0:
    CIRCLE(XO*IX,YO*IY),6
550 A=X1*IX:B=Y1*IY: PAINT (A+4,B),0
560 IF P(X1,Y1)<>0 THEN CIRCLE (A,B),3
570 EX=SGN(XO-X1): EY=SGN(YO-Y1)
580 IF RND(DZ)<=1 THEN 620
590 IF RND(2)=1 THEN IF X1+EX>0 AND
    X1+EX<=X THEN X1=X1+EX: GOTO 610
600 IF Y1+EY>0 AND Y1+EY<=Y THEN
    Y1=Y1+EY
610 A=X1*IX: B=Y1*IY
620 CIRCLE (A,B),7: PAINT (A+4,B),1
630 IF P(X1,Y1)>0 THEN P(X1,Y1)=1
640 REM *  VOCE  FOI  ATINGIDO ??      *
650 IF X1=XO AND Y1=YO THEN FOR I=1 TO
    1000:NEXT I:PRINT@ 266,"QUE PENA..":
    SOUND 1,30: GOTO 690
660 IF B(X1,Y1)<>0 THEN FOR I=1 TO
    B(X1,Y1)*40:PLAY "L200;A":NEXT I:
    DZ=DZ/B(X1,Y1): B(X1,Y1)=0
670 DZ=DZ+.1
680 GOTO 370
690 PRINT@ 325,"VOCE FEZ "T" PONTOS!!"
700 FOR ZZ=1 TO 4000:NEXT ZZ: RUN
```


CONECTE CINCO

PARA LINHA APPLE

Você vai jogar contra o computador. O programa monta a tela dividindo-a em 49 células — 7 linhas e 7 colunas. Cada jogador dispõe de uma cor para preencher as células, como se fossem peças ocupando as casas de um tabuleiro. O objetivo é colocar cinco peças da mesma cor em casas consecutivas — o que pode ser feito tanto na horizontal como na vertical. Joga um por vez, começando pela linha inferior. Indique o número da coluna em que você quer colocar a peça e tente conectar cinco. Mesmo que você não ganhe, conseguirá um belo efeito visual quando as cores varrerem a tela.




```
]LIST

10  REM  **********************
11  REM  ***                ***
12  REM  *** CONECTE CINCO ***
13  REM  ***                ***
14  REM  **********************

20  GOSUB 1000: REM  INTRUCOES
30  GOSUB 2000: REM  INICIO
40  GOSUB 3000: REM  JOGO
50  GOSUB 4000: REM  FIM
60  END
1000  REM  *** INSTRUCOES ***
1010  TEXT : NORMAL : HOME
1020  VTAB 3: HTAB 10: PRINT "***
      CONECTE 5 ***
1030  VTAB 7: PRINT "O OBJETIVO D
      O JOGO E' COLOCAR 5 DAS SUAS
       PECAS EM UMA COLUNA, TANTO
      NA HORIZONTAL COMO NA VERTIC
      AL "
1035  PRINT
1040  PRINT "QUANDO FOR SUA VEZ D
      E MOVER, ENTRE O NUMERO DA C
      OLUNA QUE DESEJA OCUPAR COM
      SUA PECA"
1045  PRINT
```